

AFRL-IF-RS-TR-1998-142
Final Technical Report
July 1998



KNOWLEDGE-BASED USER-COMPUTER INTERFACE DESIGN, PROTOTYPING AND EVALUATION - THE DESIGN PRO ADVISORY SYSTEM

Drexel University

Stephen J. Andriole, Charlton A. Monsanto, and Lee Scott Ehrhart

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

19980911 002

**AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK**

DTIC QUALITY INSPECTED 1

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

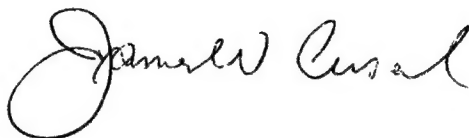
AFRL-IF-RS-TR-1998-142 has been reviewed and is approved for publication.

APPROVED:



ROBERT M. FLO
Project Engineer

FOR THE DIRECTOR:



JAMES W. CUSACK
Chief, Information Systems Division
Information Directorate

If your address has changed or if you wish to be removed from the Air Force Research Laboratory Rome Research Site mailing list, or if the addressee is no longer employed by your organization, please notify AFRL/IFSA, 525 Brooks Road, Rome, NY 13441-4505. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE July 1998	3. REPORT TYPE AND DATES COVERED Final Aug 92 - Sep 95		
4. TITLE AND SUBTITLE KNOWLEDGE-BASED USER-COMPUTER INTERFACE DESIGN, PROTOTYPING AND EVALUATION - THE DESIGN PRO ADVISORY SYSTEM		5. FUNDING NUMBERS C - F30602-92-C-0119 PE - 62702F PR - 5581 TA - 32 WU - 10		
6. AUTHOR(S) Stephen J. Andriole, Charlton A. Monsanto, and Lee Scott Ehrhart				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Drexel University College of Information Science & Technology Center for Multidisciplinary Information Systems 3141 Chestnut Street Philadelphia PA 19104-2875		8. PERFORMING ORGANIZATION REPORT NUMBER N/A		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory/IFSA 525 Brooks Road Rome NY 13441-4505		10. SPONSORING/MONITORING AGENCY REPORT NUMBER AFRL-IF-RS-TR-1998-142		
11. SUPPLEMENTARY NOTES Air Force Research Laboratory Project Engineer: Robert M. Flo/ISFA/(315) 330-2334				
12a. DISTRIBUTION AVAILABILITY STATEMENT Approved for public release; distribution unlimited.		12b. DISTRIBUTION CODE		
13. ABSTRACT (Maximum 200 words) This report details the development of the DesignPro interactive computer-based advisory system for user-computer interface (UCI) design, prototyping and evaluation. DesignPro permits designers of user computer interfaces to represent requirements, to build prototypes, and to evaluate their impact -- all via a "workbench" of user accessible functions. DesignPro supports the UCI designer; it does not call for the replacement of human UCI expertise in the design process. The methodology assumes that commercial-off-the-shelf (COTS) software can be used to create an integrated environment for designing, prototyping and evaluating interactive user computer interaction routines. The overall process includes interaction among knowledge templates to develop a requirements model that, in turn, helps yield displays and UCI routines which, in turn, suggest a prototyping strategy which, in turn, identifies evaluation tactics. Anchored in the systems engineering approach to interactive systems design and development, an initial prototype of DesignPro was released in January 1993; refinements were made to the prototype with a final prototype released in August 1995. The prototypes were used to validate workstation requirements and to communicate what the system does, as well as permitted the integration of concepts, tools, and COTS software programs into the design.				
14. SUBJECT TERMS Knowledge-Based, User Computer Interface, Prototype, Human Computer Interaction		15. NUMBER OF PAGES 146		
		16. PRICE CODE		
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL	

Table of Contents

Executive Summary

Acknowledgements

1.0 Introduction

1.1 Information & Software Systems Engineering

- 1.1.1 Systems Engineering Goals*
- 1.1.2 System Definition*
- 1.1.3 Design & Development*
- 1.1.4 Systems Engineering Management*
- 1.1.5 Systems Engineering & DesignPro*

1.2 The Primacy of Requirements, Prototyping & Evaluation

1.3 User-Computer Interface Design, Prototyping & Evaluation

1.4 Project Overview

1.5 Project Results

- 1.5.1 Knowledge-Based UCI Design, Prototyping & Evaluation*
- 1.5.2 The FLEX Case Study*
- 1.5.3 The Cognitive Task Analysis Process*
- 1.5.4 The Knowledge-Based UCI Design, Prototyping & Evaluation Workbench*

2.0 UCI Design, Prototyping & Evaluation

2.1 UCI Requirements Analysis

2.2 UCI Features

2.3 UCI Prototyping

2.4 UCI Evaluation

2.5 Knowledge-Based UCI Design, Prototyping & Evaluation

3.0 The FLEX Case: Feature Enhancements Via Prototyping

3.1 Defining the Problem

3.2 Identifying & Modeling the Tanker Duty Officer's Cognitive Tasks Requirements

3.2.1 Defining the FLEX Environmental/Situational Context

3.2.2 Profiling the FLEX Organizational/Doctrinal Context

3.2.3 Profiling the Tanker Duty Officer (TDO)

3.2.4 Profiling the TDO's Functional Tasks

3.2.4.1 Task Output

3.2.4.2 Task Response

3.2.4.3 Procedures & Sub-Tasks

- 3.2.4.4 Task Input
- 3.2.4.5 Task Feedback

3.2.5 Profiling the TDO's Decision-Making Tasks

- 3.2.5.1 Stimulus - Characteristics of the Situational Context & Data Inputs
- 3.2.5.2 Hypothesis - Situation Assessment Task Characteristics
- 3.2.5.3 Option - Course of Action Decision Tasks
- 3.2.5.4 Response - Planning, Coordination & Execution of Decisions

3.2.6 The FLEX Cognitive Task Requirements (CTRs)

3.2.7 Specific Cognitive Task Requirements

3.3 Integrating Cognitive Task Requirements into the Systems Requirement Document

3.4 Transitioning Requirements to a UCI Design Concept

3.5 Developing an Interactive Prototype of the UCI Design Concept

3.6 Evaluating the UCI Design Concept

3.6.1 Developing Evaluation Goals

3.6.2 Selecting Evaluation Methods

3.7 The Cognitive Task Analysis Handbook

4.0 *The UCI Design, Prototyping & Evaluation Workbench*

4.1 Hardware/Software Configuration

4.2 Operation

4.3 The FLEX Tutorial

5.0 *Summary & Conclusions*

6.0 *References & Bibliography*

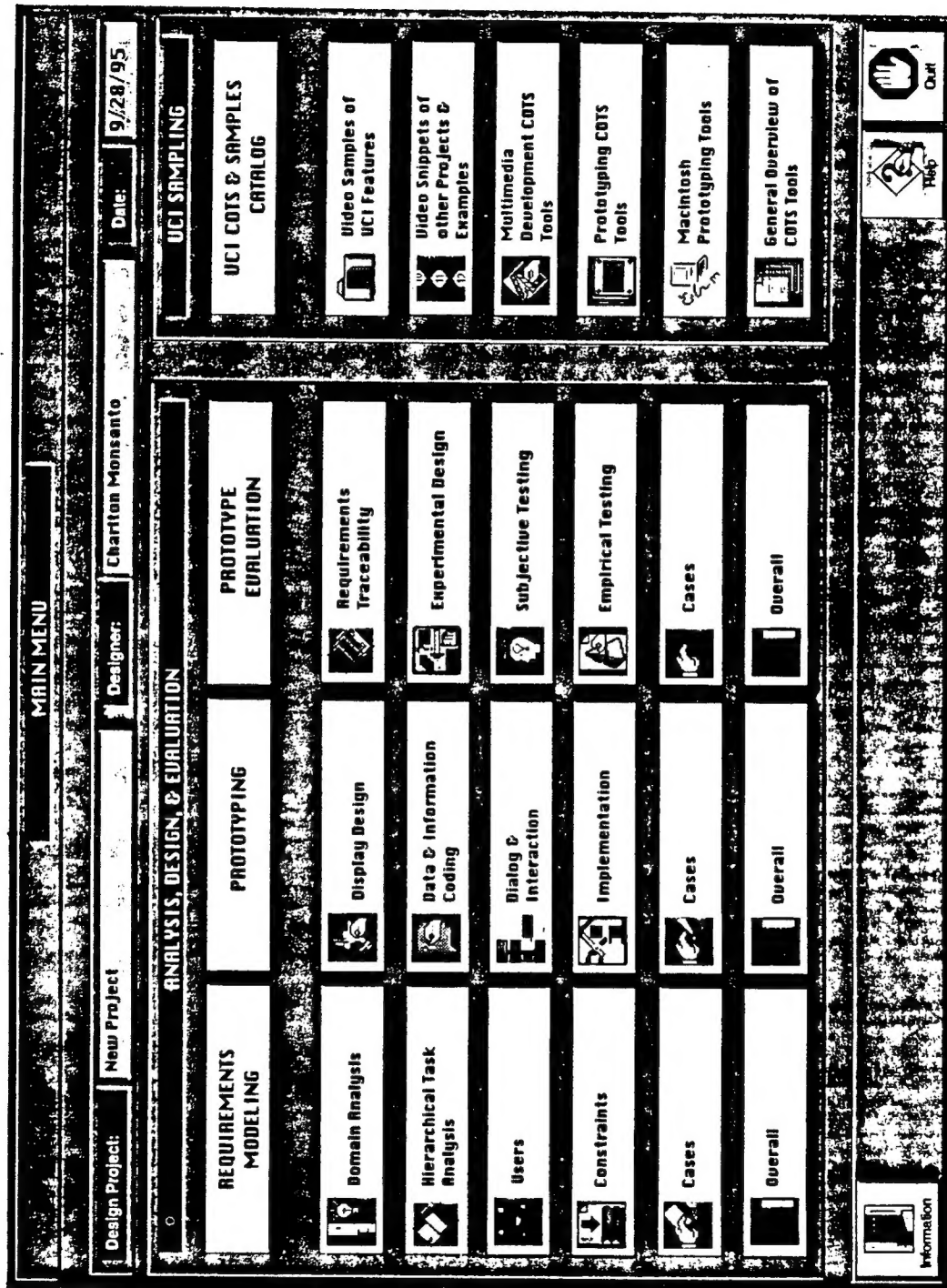
Appendices

Appendix A: Screen Displays from Workbench

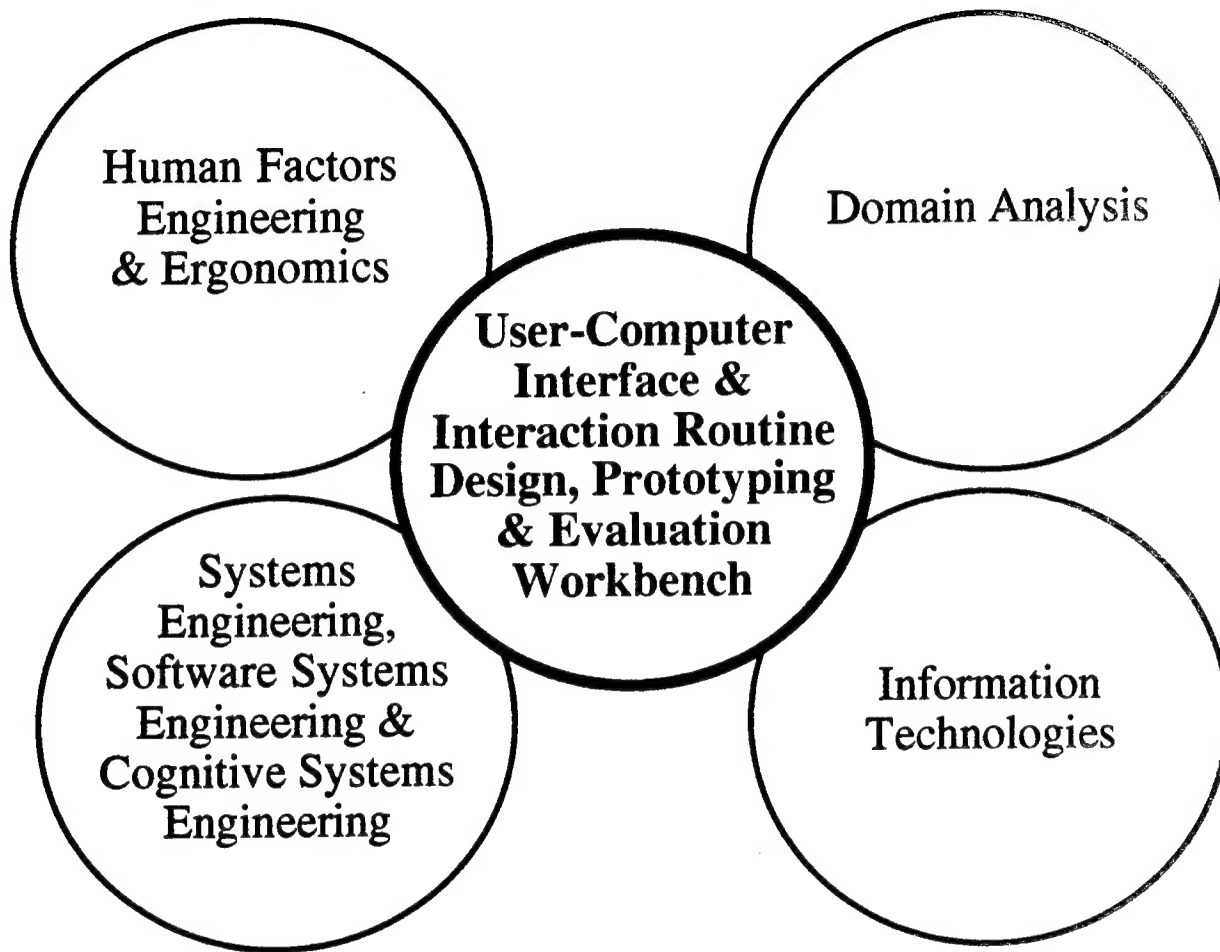
Appendix B: FLEX "Before" & "After" Screens

Executive Summary

- *This final technical report covers the period from August 1, 1992 through August 31, 1995. The report describes progress made in the development of the DesignPro interactive computer-based advisory system for user-computer interface (UCI) design, prototyping and evaluation.*
- *The overall process includes interaction among knowledge templates to develop a requirements model that, in turn, helps yield displays and UCI routines which, in turn, suggest a prototyping strategy which, in turn, identifies evaluation tactics.*
- *The DesignPro system supports to the UCI designer; it does not call for the replacement of human UCI expertise in the design process. The methodology assumes that commercial-off-the-shelf (COTS) software can be used to create (simulate) an integrated environment for designing, prototyping and evaluating interactive user computer interaction routines.*
- *The project was anchored in the systems engineering approach to interactive systems design and development; the throwaway ---> evolutionary prototyping approach to validate requirements was implemented. An initial prototype was released in January of 1993; another in April of 1993 and another in October 1993; refinements were made to the prototype in January and March of 1994 and then again in April 1995, with a final prototype release in August 1995. The prototypes were used to validate workstation requirements and to communicate what the system does. They also permitted the integration of concepts, tools, & COTS software programs into the design.*
- *The project has also pursued a case study within its scope. The FLEX case study was completed during this reporting period and presented to Rome Laboratory personnel in July 1994. FLEX illustrated how the UCI design, prototyping, and evaluation methodology embedded in DesignPro can be used to design, prototype and evaluate varieties of command and control interfaces.*
- *The DesignPro Advisory System permits designers of user computer interfaces to represent requirements, to build prototypes, and to evaluate their impact -- all via a "workbench" of user accessible functions.*
- *The following figure presents the DesignPro workbench. Note the major functional areas and the system's ability to show examples of the features that comprise user computer interfaces as well as examples of off-the-shelf prototyping tools via a "browser" capability.*



- *The project synthesized findings from a variety of sources and disciplines -- as suggested by the following figure:*



The Project's Analytical Backdrop

- *The project's ultimate payoff will depend upon the nature of the user-computer interface design applications to which the workbench is applied; a large number of applications will provide insight into the operational capabilities of the interface and the analytical and design assumptions upon which it is based.*
- *Ultimately the workbench demonstrates a growing trend in the design arena: the embedding of more and more design expertise in the knowledge-based software systems capable of -- in most cases -- advising designers and -- in a few cases -- automating design processes.*

Acknowledgments

There are many professionals to which we owe thanks. Lt. Col. Tom Triscari, Jr. (USAF) was the catalyst that got the process going. Dick Slavinski took the initiative and made it happen, while Rob Flo managed us along the way.

We'd also like to thank the students at Drexel University who supported the project and the University itself which subsidized the research and development process in several important ways. Of special administrative and technical help along the way was Betty Jo Hibberd, to which we also owe much thanks.

Bill Ringle implemented much of the system -- and cleverly figured out how to get a rather large knowledge base firing at the right time and in the right places.

The FLEX design team helped tremendously. Tom Clark, Earl Lebat, Gerald Ruigrok, Georgeanne DeWalder, Capt. Scott Bourgeois (USAF), Lt. Keith Felter (USAF), and the senior advisors from the FLEX Working Group. Thanks also to Capt. Gary Stefanich (USAF) from the APS Program Office, who helped to develop the scenarios for the experimental study. Col. Larry Simmons was instrumental in arranging for subjects and domain experts from the 509th Air Refueling Squadron.

George Mason University supported and administered the project while Lee Ehrhart was at GMU during the first half of the project.

We'd like to thank all of the above professionals and organizations -- along with the Department of the Air Force -- for their support and encouragement.

1.0 Introduction

The last decade has seen the proliferation of systems that are highly user-computer interaction-intensive. Yet overall performance has not justified the investments we have made in tactical "decision aids," support systems, and larger information systems. We still hear complaints about how difficult systems are to use, how they provide only the barest analytical support, and how they -- ultimately -- fail to satisfy the most important requirements.

The response to these and related problems has been incremental and disembodied. Perhaps most importantly, we have failed to leverage advanced information technology in our systems design. We have failed to use technology to help define requirements, to find the right analytical methods, or to enhance user-computer interfaces.

We have not invested nearly enough in low-cost design environments, environments that permit the rapid testing and evaluation of ideas -- regardless of how controversial they might seem. Bureaucratic, administrative and financial disincentives surround the design and development process: it is easier to pursue incremental fixes than to challenge whole design philosophies. Low-cost design, development and testing environments would permit assessments about what investments make sense and what should be avoided.

With these problems in mind, we developed a workstation-based "environment" for designing, developing (prototyping) and testing new interaction concepts quickly and cost-effectively. The environment relies upon the design principles anchored in the discipline of systems engineering, commercial-off-the-shelf (COTS) software, and lots of examples of user computer interaction "features" and examples of user computer interfaces.

In order to design and develop the kind of system capable of supporting the design, prototyping and evaluation of simple and complex user-computer interfaces (UCIs), it was necessary to adopt an overarching design and development methodology, a methodology that could (a) guide the design and development of DesignPro and (2) be appropriately embedded with DesignPro itself. As suggested briefly above, information and software systems engineering provided this disciplinary framework.

Figure 1 describes the project's analytical backdrop:

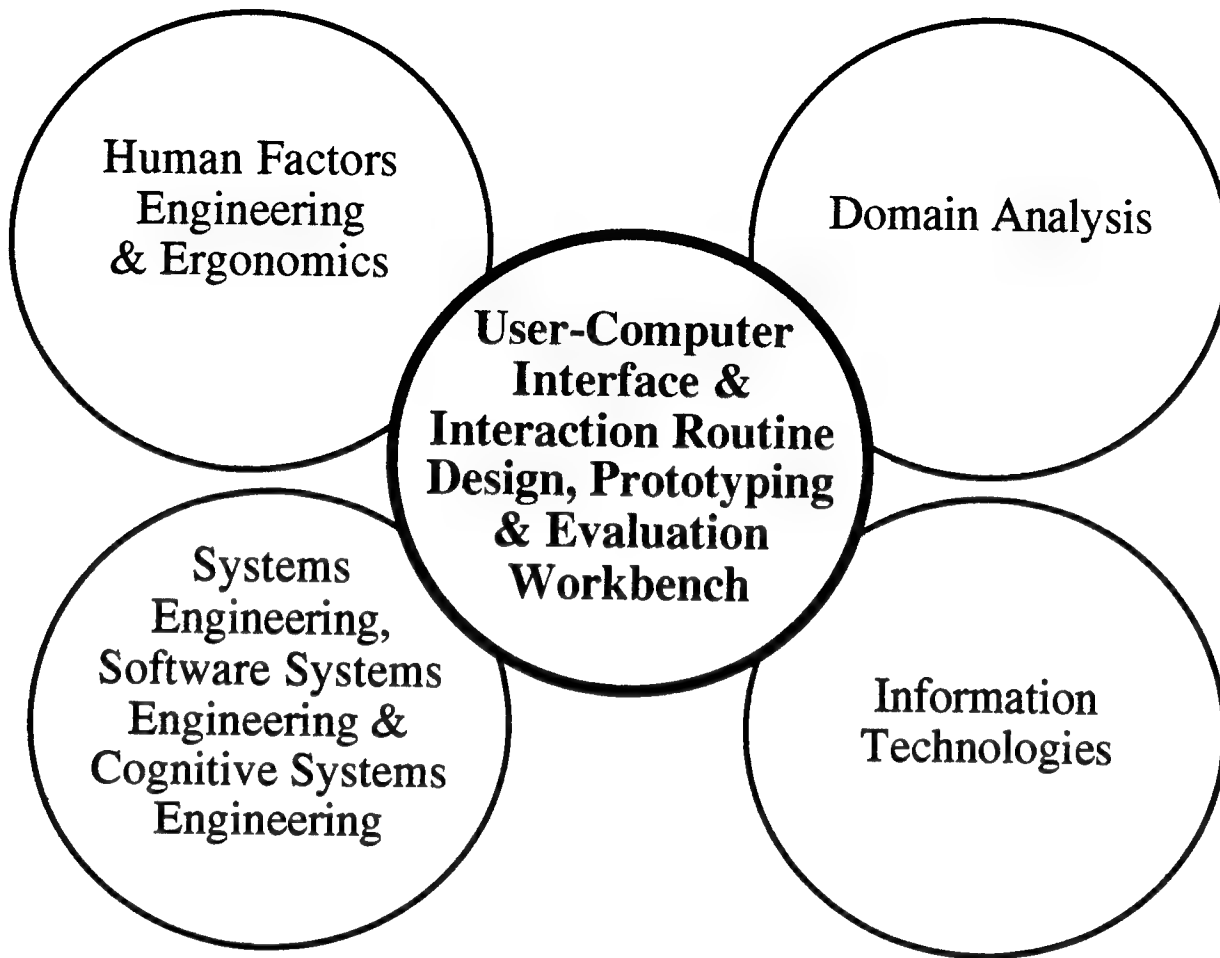


Figure 1: Project Backdrop

1.1 Information & Software Systems Engineering

Systems engineering has had the benefit of many models, life cycles, processes and methods honed over the years into a comprehensive methodology for the design, development, testing, and maintenance of systems of all kinds. While the emphasis here is on the systems engineering of software-intensive systems, the discipline supports systems design and development regardless of domain, regardless of organizational context.

Systems engineering traces its origins to the work of Hall (1962), Chestnut (1965; 1967), Chase (1974), Sage (1977; 1983), and Blanchard and Fabrycky (1981; 1990). More recently, Sage (1992), Blanchard (1991), Eisner (1988), and Chapman, Bayhill and Wymore (1992) have refined the methods, tools techniques, activities, functions and purpose of systems engineering.

Along the way an overarching Department of Defense standard evolved -- MIL-STD 499A -- which describes the process as follows:

"Systems engineering is the . . . logical sequence of activities and decisions transforming an operational need into a description of system performance parameters and a preferred system configuration ... systems engineering is the application of scientific and engineering efforts to (a) transform operational need into a description of system performance parameters and a system configuration through the use of an iterating process of definition, synthesis, analysis, design, test, and evaluation; (b) integrate related technical parameters and ensure compatibility of all physical, functional, and program interfaces in a manner that optimizes the total system definition and design; (c) integrate reliability, maintainability, safety, survivability, human, and other such factors into the total engineering effort to meet cost, schedule, and technical performance objectives."

The revised standard (DOD, 1992) describes the systems engineering process as follows:

"A comprehensive, iterative problem solving process that is used to: (a) transform validated customer needs and requirements into a life-cycle balanced solution set of system product and process designs, (b) generate information for decision-makers, and (c) provide information for the next acquisition phase."

Sage (1992) offers three definitions of systems engineering which constitute perspectives on the process. He defines systems engineering *structurally*, *functionally*, and in terms of *purpose*. Structurally, Sage sees systems engineering as a "management technology to assist clients through the formulation, analysis, and interpretation of the impacts of proposed policies, controls or complete systems upon the perceived needs, values, institutional transactions of stakeholders."

Functionally, systems engineering -- according to Sage (1992) -- is an "appropriate combination of theories and tools, carried out through the use of a suitable methodology and set of systems management procedures."

Sage's purposeful definition of systems engineering assumes that the "purpose of systems engineering is information and knowledge organization that will assist clients who desire to develop policies for management, direction, control and regulation activities relative to forecasting planning, development, production and operation of total systems to maintain overall integrity and integration as related to performance and reliability."

Eisner (1988) defines systems engineering as “an iterative process of top-down synthesis, development, and operation of a real-world system that satisfies, in a near optimal manner, the full range of requirements for the system.” Eisner (1988) describes the systems engineering process as consisting of the following “elements:”

1. *Requirements analysis;*
2. *Requirements allocation;*
3. *Functional analysis;*
4. *Functional allocation;*
5. *Specification analysis;*
6. *Specification development;*
7. *Preliminary design;*
8. *Interface definition;*
9. *Schedule development;*
10. *Preliminary cost-analysis;*
11. *Technical performance measurement;*
12. *Trade-off/alternative analysis;*
13. *Pre-planned product improvement;*
14. *Final design;*
15. *Schedule update;*
16. *Cost update;*
17. *Fabrication;*
18. *Coding;*
19. *Preliminary testing;*
20. *Debugging & reconfiguration;*
21. *Testing & integration;*
22. *Updates:*
 - A. *Schedule;*
 - B. *Cost;*
 - C. *Technical performance measurement;*
23. *Documentation;*
24. *Training; and*
25. *Production.*

Eisner's generic systems engineering process appears in Figure 2. Note the emphasis on “front-end” activities, activities that determine development and maintenance requirements. The importance that systems engineers place on this front-end process is precisely the emphasis that defines the purpose, embedded processes and outcome of our knowledge-based advisory workbench -- DesignPro.

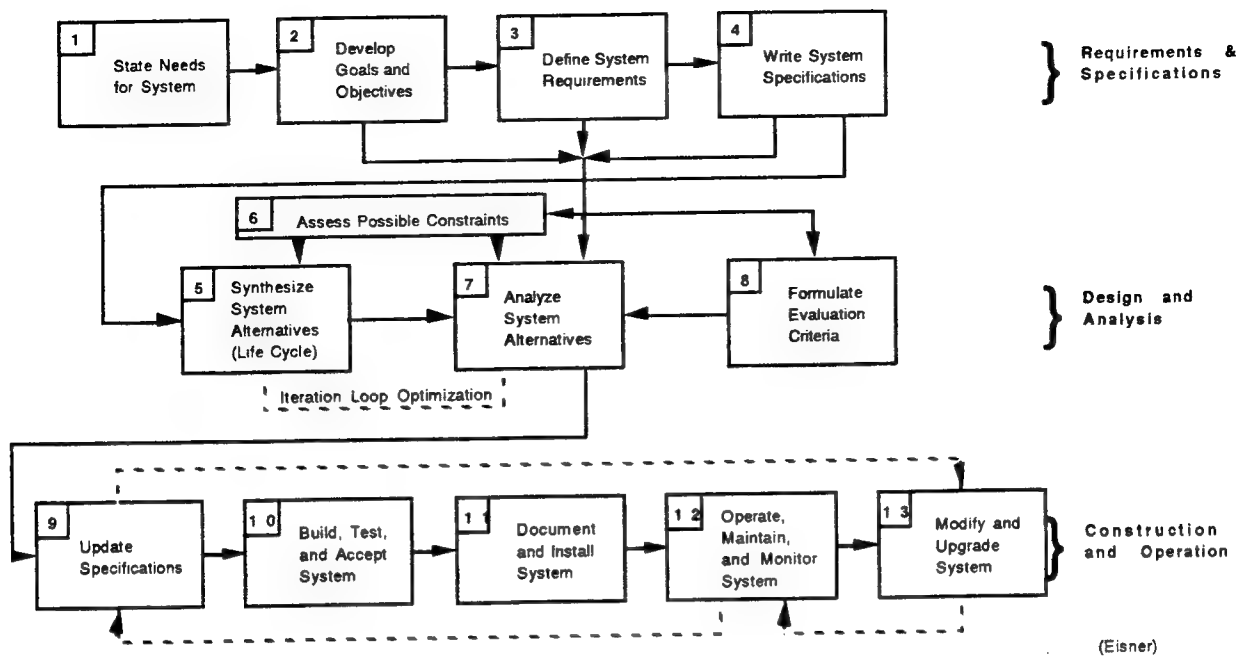


Figure 2: Eisner's Generic Systems Engineering Process

Blanchard's (1991) systems engineering process combines these and other phases, steps and elements into a process that we can use to link to various alternative systems engineering process models and life cycles.

1.1.1 Systems Engineering Goals

The software systems engineering process is implemented in order to achieve certain objectives. Sage (1992) lists the following:

- *All (life cycle) encompassing*
- *Problem understanding*
- *Communication*
- *Early capture of design & implementation needs*
- *Bottom-up & top-down design & development*
- *Alternative systems management approaches*
- *Process & product quality assurance*
- *Product evolution*
- *Support for configuration management standards*
- *Support for automated design & development aids*
- *Teachable & transferable methodology*
- *All phase definition & documentation*
- *Operational product functionality, revisability & transitioning*
- *Support system product development & system user organizations*

The thrust of this list is that successful systems engineering should be disciplined, structured, informed by knowledge and data, repeatable, documented, and well managed. In effect, the argument here is for success through process maturity. But perhaps more importantly, Sage's definitions of systems engineering speak directly to objectives and goals. In other words, the objectives of the systems engineering process include attention to the *structure*, *function*, and *purpose* of systems engineering.

1.1.2 System Definition

The essence of the systems engineering process is requirements ---> design ---> development efficiency. The primacy of requirements is well documented (Andriole, 1989, 1990; Sage, 1992; Sage & Palmer, 1990; Davis, 1989, 1992). Eisner's process model serves us well again here. It focuses directly on the "definition of need" and the development of a preliminary design via requirements analysis, modeling, and allocation, trade-off analysis, and input to the detailed design and development process.

Mainstay requirements tools include functional block and flow diagrams which identify, define and communicate the functions, tasks and sub-tasks that the system-to-be should perform. The specification of a system's external behavior is endemic to the overall systems engineering process; in fact, external behavior specification is a filter through which subsequent design and development decisions should be passed. From another perspective, external behavior specification is a gate that cannot be traversed until consensus about functional capabilities emerges. Other tools include NxN charting, hierarchical task models, and data/control flow models (Sailor, 1990; Eisner, 1988).

In addition to these tools are a variety of others. Eisner (1989) describes dependency diagrams, signal-flow block diagrams, Petri Nets, hierarchical input-process-output diagrams, Warnier-Orr diagrams, Michael Jackson diagrams, action diagrams, sequence and timing diagrams, parameter dependency diagrams, logic flow charts, Nassi-Shneiderman charts, and decision-network diagrams, among others. These and related methods, tools and techniques -- many of which are computer-based -- permit systems engineers to define, model, and validate requirements (prior to expensive design or development activities).

It is essential to remember that systems engineers expect conflict during the requirements process. They expect requirements to evolve over time as they are discovered by the systems analysis team. But they also expect a prioritized requirements hierarchy to emerge early on in the process.

Risk also becomes a major issue during the front-end requirements analysis, modeling and prototyping process. It is, in practice, another side of trade-off analysis, where stakeholders are asked to prioritize and then assess feasibility. If a high important requirement cannot easily be satisfied then risk rises dramatically. Risk also rises when technological feasibility is assessed "low." It is not unusual, for example, for system plans to call for the insertion of a specific technological capability at some point in the systems life cycle. But what if the capability does not exist? How serious might delays become? Should investments be made today to engineer a capability tomorrow? Risk assessment is essential to system success, since early miscalculations almost always cost a great deal of time and money downstream.

The key to the process is early diagnosis of key requirements, feasibility, and risk. When the results of these processes suggest conflict then trade-off analysis ensues. Here the systems engineer calculates -- via empirical and subjective data -- the relationships among competing requirements and the cost-benefit calculus derived from the relationships. Where requirements prioritization can yield a "benefit-only" list of requirements, trade-off analysis should yield a cost-benefit-based list (where *cost* is defined as time, money and talent).

Good systems engineering requires that system requirements be identified, modeled, validated, discovered, and "traced" throughout a system's life. Data regarding the source of requirements, their original form, their relative importance, their flexibility, and their precise location in prototypes all must be stored for easy access to the systems engineering team.

All of this front-end work is intended to deepen our understanding of initial requirements, to prevent major requirements from falling through project cracks, and to provide insight into the next -- prototyping -- phase of the front-end systems engineering process.

Requirements are thus validated with reference to constraints and alternative designs. Trade-off analysis, risk analysis and other forms of "sensitivity analysis" are conducted to determine how diagnostic requirements are, can realistically be, and might be given additional investments.

The functional outcome of this process is a system synthesis and definition of sufficient diagnosticity to permit more detailed design and development.

1.1.3 Design & Development

The detailed design and development process (a) assumes that the system

definition that emerges from the front-end process is sound and (b) that the steps toward production are clear. System/product design, prototype development, and prototype testing & evaluation constitute the detailed design and development process.

The design and development process can be accelerated via prototyping. But prototyping is most effective when requirements are reasonably well understood, system trades have been performed, and a system concept likely (but obviously not certain) to satisfy requirements has emerged. There are several forms worth noting: throwaway, evolutionary and hybrid prototyping. Throwaways are used for new system concepts, while evolutionary prototypes are used when requirements are much better understood or when a well-documented system is under-going enhancement. Hybrids are used when parts are well understood and some parts are not.

Systems engineers prototype routinely; they also evaluate the prototypes to determine the extent to which they satisfy functional requirements as well as specific non-functional requirements, such as security, modifiability, and transportability. Dorfman (1990), among many others, suggests that prototyping should precede systems and software design, that the performance and evaluation of the prototype should inform the design and development process. Prototypes can -- and should -- be evaluated with reference to their usability, the degree to which they satisfy functional and non-functional requirements, and likely implementation costs.

1.1.4 Systems Engineering Management

The systems engineering process must be carefully managed. By definition the process is complex and management quickly becomes essential as the phases of the life cycle are implemented.

Eisner (1989) assumes that at the highest level the systems engineering process itself must be managed, that engineering speciality areas must be identified and managed, and that technical program planning and control is essential. Sage (1992) expands the concept to include organizational management and structures, quality assurance via configuration management, reviews and standards, and *strategic* quality assurance and management.

Blanchard (1991) suggests that systems engineering management can be defined around the systems acquisition process and the major milestones that track with the major phases of the systems engineering process. Blanchard's management model identifies the need for a formal systems engineering management plan

(SEMP), a test and evaluation (T&E) master plan (TEMP), and various design reviews. These plans and reviews are incarnated in "specifications," such as the venerable " 'A' spec."

In addition, systems engineering management involves decision-making that occurs as a result of feedback from specific kinds of analyses, such as risk analysis, technology assessment, and the like. Systems engineering management ideally locates "off-ramps" in the design and development process, off-ramps that inform -- and sometimes delay or cease -- the design, development and production process. The ideal management process is one that is analytical, adaptive and decisive.

Sage (1992) suggests that systems engineering management can be defined around audits, reviews, standards, and systems integration -- all with reference to quality.

All of these management concepts and objectives travel with the systems engineering process itself. Structurally, systems engineering is -- according to Sage -- technology management. The Department of Defense Standards all find it difficult to separate management from technology and engineering.

1.1.5 Systems Engineering & DesignPro

It is important to note, however, that our view of systems engineering is not without application focus. We are concerned primarily with the systems engineering of *software-intensive systems*. We are also concerned with the systems engineering of individual software-intensive products and product lines, and as an overarching organizational planning and management process.

The software systems engineering life cycle adopted here appears below. It was adapted from the generic systems engineering process:

1. Analyze Requirements

- A. Method(s): Interview, Observe, Simulate ...
- B. Tool(s): Off-The-Shelf Software: Outline/Idea Processors ...
- C. Objective: First Cut at User Requirements; Document for Continued Use ...

2. Model Requirements

- A. Method(s): Hierarchical Decomposition, IPO, Causal, Cognitive Maps/Mental Models, Simulations ...

- B. Tool(s): OTS Software: Hierarchical Tools, Simulation Tools, General Purpose Modeling Tools ...
- C. Objective: Initial Transition from Observational & Textual Data & Information into Structured, Organized Model of User Requirements ...

3. Assess Constraints

- A. Methods(s): Cost-Benefit Analysis, Impact Assessments, Multi-Attribute Utility Assessment (MAUA), Qualitative & Quantitative Modeling ...
- B. Tool(s): OTS Software: Decision Analytic Packages, MAUA Tools, Modeling Packages ...
- C. Objective: Identify Key Obstacles to Development & Implementation; Measure the Impact with Reference to Requirements ...

4. Prioritize Requirements

- A. Methods(s): Qualitative & Quantitative Rank-Ordering; Simulation-Based Prioritization ...
- B. Tool(s): OTS Software: Hierarchical Decomposition Packages; MAUA Packages ...
- C. Objective: Initial List of Most Important Requirements Given Constraints & Initial Requirements Model ...

5. Develop Alternative System Concepts

- A. Methods(s): Cost-Benefit Analysis, Impact Assessments, Requirements Tracing ...
- B. Tool(s): OTS Software: Decision Analytic Packages, Simple Configuration Models ...
- C. Objective: Identify System Concepts Most Likely to Satisfy Prioritized Requirements ...

6. Trade-Off Analysis

- A. Methods(s): Cost-Benefit Analysis, Impact Assessments, Multi-Attribute Utility Assessment (MAUA) ...
- B. Tool(s): OTS Software: Decision Analytic Packages, MAUA Tools ...
- C. Objective: Identify Key Obstacles to Development & Implementation; Measure the Impact with Reference to Requirements ...

7. System Concept Specification

- A. Method(s): Cost-Benefit Analysis, Impact Assessments, Requirements Tracing, Sizing ...
- B. Tool(s): OTS Software: Decision Analytic Packages, Simple Configuration Models, Process Models ...
- C. Objective: Identify System Concepts Most Likely to Satisfy Prioritized Requirements ...

8. Prototype Requirements

- A. Method(s): Throwaway/Evolutionary: Screens, Interactions, Output (External Behavior of System) ...
- B. Tool(s): OTS Software: Screen Formatters, Interactive Prototypers, Code Generators, UIMSs ...
- C. Objective: Throwaway Prototype: Convert Requirements Model into a System Concept that Demonstrates UCI and Overall Functionality; Evolutionary Prototype: Begin Transition to Full Scale Production (Build a Little, Test a Little) ...

9. Specify Software Requirements

- A. Method(s): DFDs, ERDs, NSs, GS, DeMarcos ...
- B. Tool(s): OTS Software: Conventional CASE Tools ...
- C. Objective: Represent the Software Design Via Consistent Notation to "Guide" Production of Code & Document the Conversion Process ...

10. Design & Develop Software

- A. Method(s): Structured Programming, Programming "Conventions" ...
- B. Tool(s): Programmer Workbenches; CASE Tools; Programming "Environments" (e.g., Ada) ...
- C. Objective: Software that Works ...

11. Test Software

- A. Method(s): Qualitative & Quantitative Methods ...
- B. Tool(s): CASE Tools; Unit Testing Tools; Diagnostic Tools ...
- C. Objective: Verified & Validated Software ...

12. Field System ...

1.2 The Primacy of Requirements, Prototyping & Evaluation

The whole project was anchored in systems engineering which itself is anchored in the primacy of requirements. Good systems engineers place requirements at front and center of the design process. Our project did precisely the same thing: we assumed that the best user computer interface would be one informed by user requirements, requirements that could be validated via prototyping. In order to determine the likely impact of specific UCI features, the project (and its DesignPro workbench) also supports evaluation, which is important in our on-going effort to prevent the programming of interfaces unlikely to enhance human-computer interaction.

1.3 User-Computer Interface Design, Prototyping & Evaluation

The project assumed that it was possible to identify the steps that together comprise the UCI design, prototyping and evaluation process. We further assumed that it was possible to model this process via the synthesis of the systems engineering process and knowledge bases placed at various locations in the design process. The simple concept appears in Figure 3:

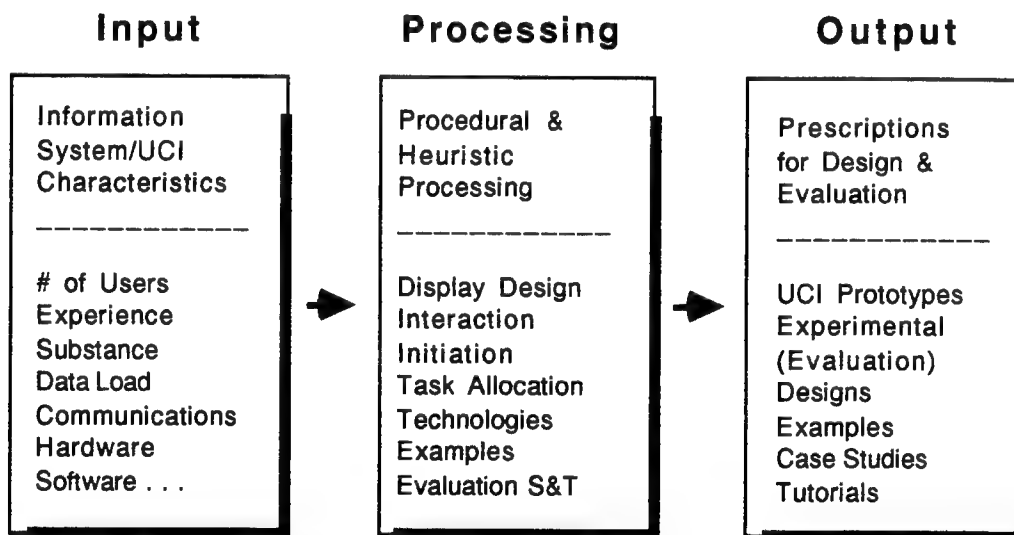


Figure 3: Simple Interaction Process

A more detailed view appears in Figure 4. This view indicates how requirements can be modeled and then “converted” via interaction with a knowledge base (comprised of simple rules about the relationships among task requirements and

UCI features) into UCI feature recommendations which, in turn, can be prototyped for subsequent evaluation.

The evaluation process sheds light onto the prototyping process. If the prototype inaccurately reflects user requirements then the evaluation will yield inconclusive or negative results; on the other hand, if the prototype enhances human computer performance, then the requirements have probably been faithfully represented throughout the design process.

Figure 4 illustrates graphically what the process looks like:

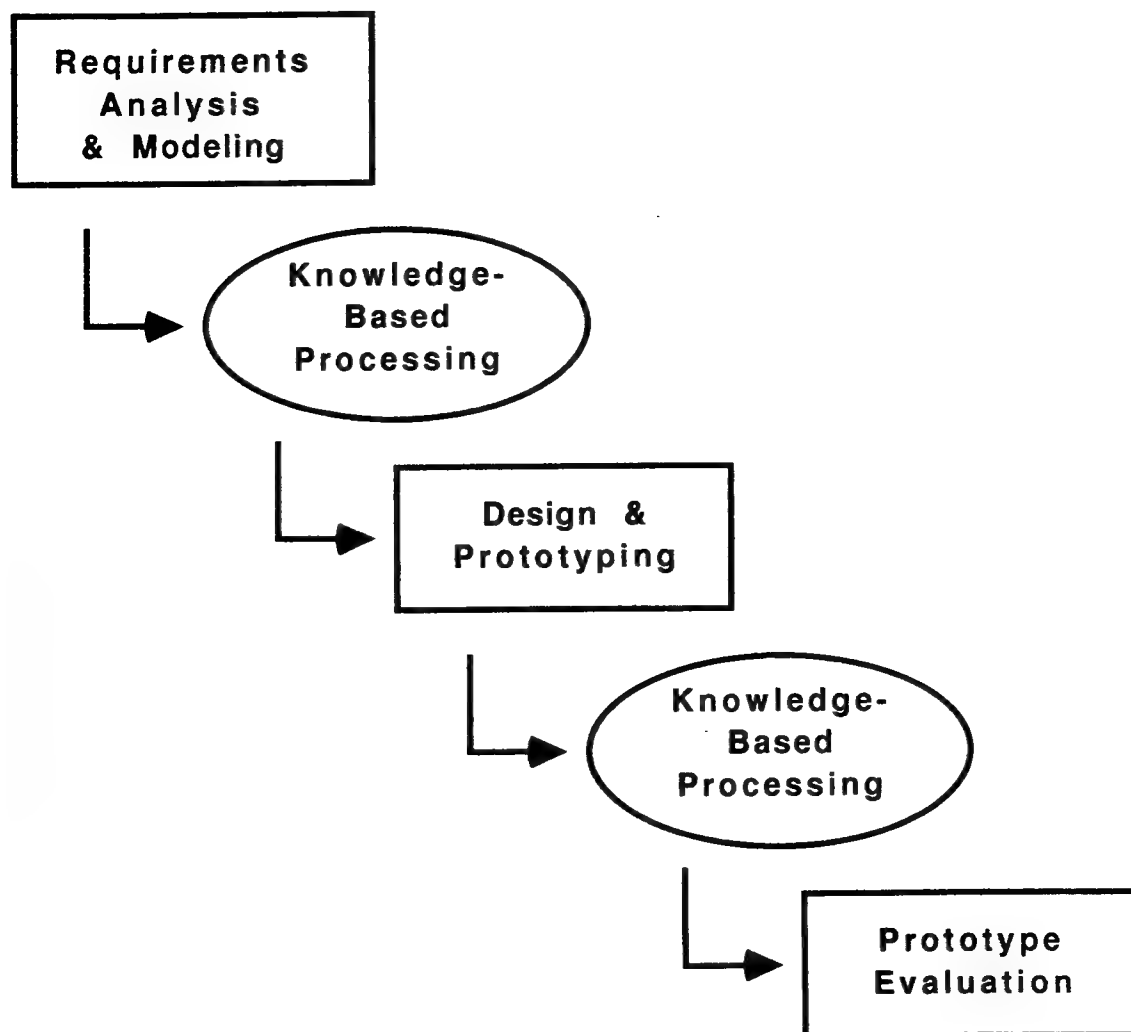


Figure 4: The Knowledge-Based Requirements ---> Prototyping ---> Evaluation Process

1.4 Project Overview

The project to incarnate UCI design, prototyping and evaluation knowledge and experience (via examples and COTS tools), began in 1992. The project was completed in the summer of 1995. Project tasks are identified below in Figure 5.

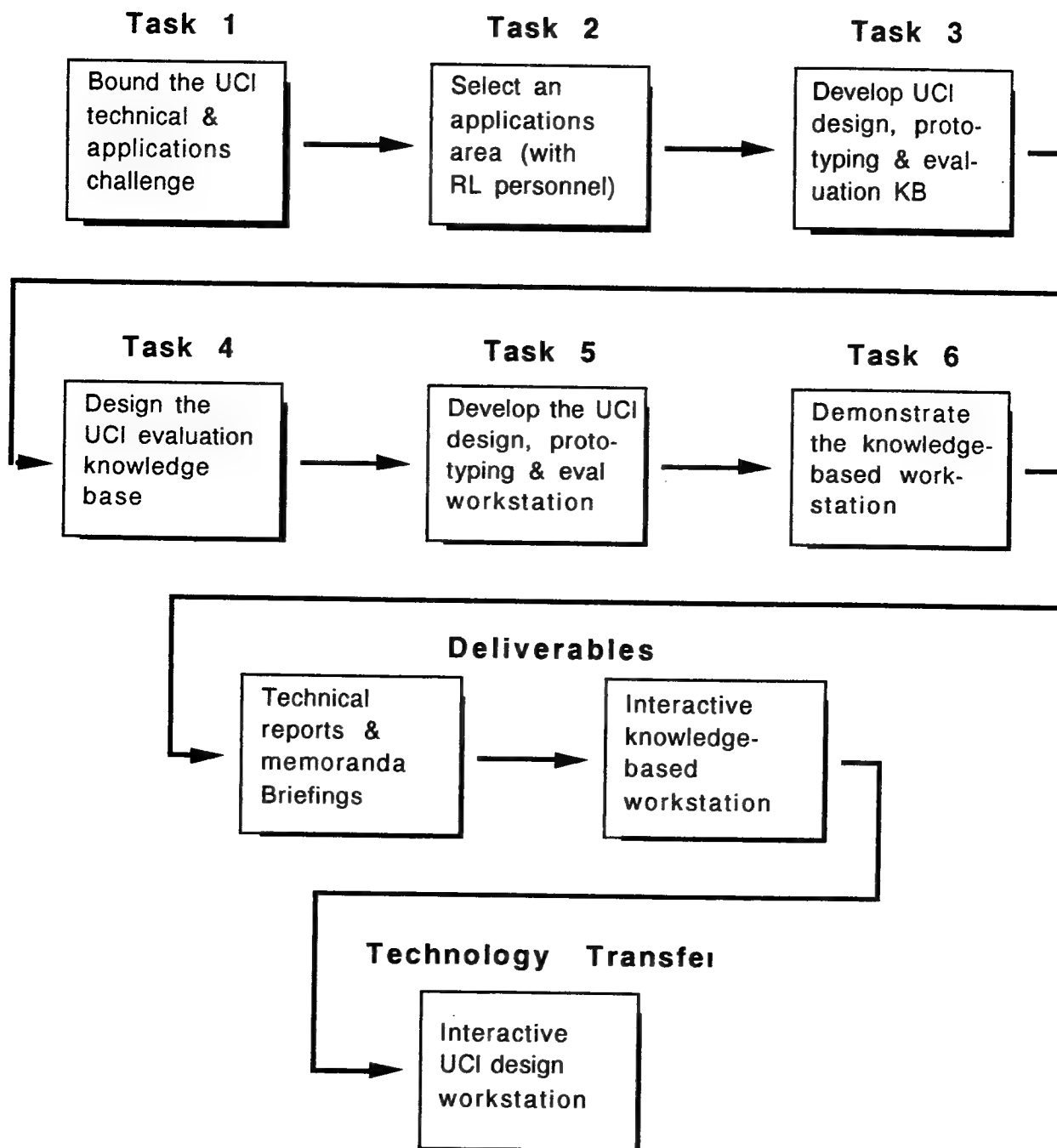


Figure 5: Project Tasks

1.5 Project Results

A prototype -- known as DesignPro -- was developed, a major case study was undertaken, a new methodology was refined, and the project was documented.

1.5.1 Knowledge-Based UCI Design, Prototyping & Evaluation

The objective of the project was the representation of the systems engineering

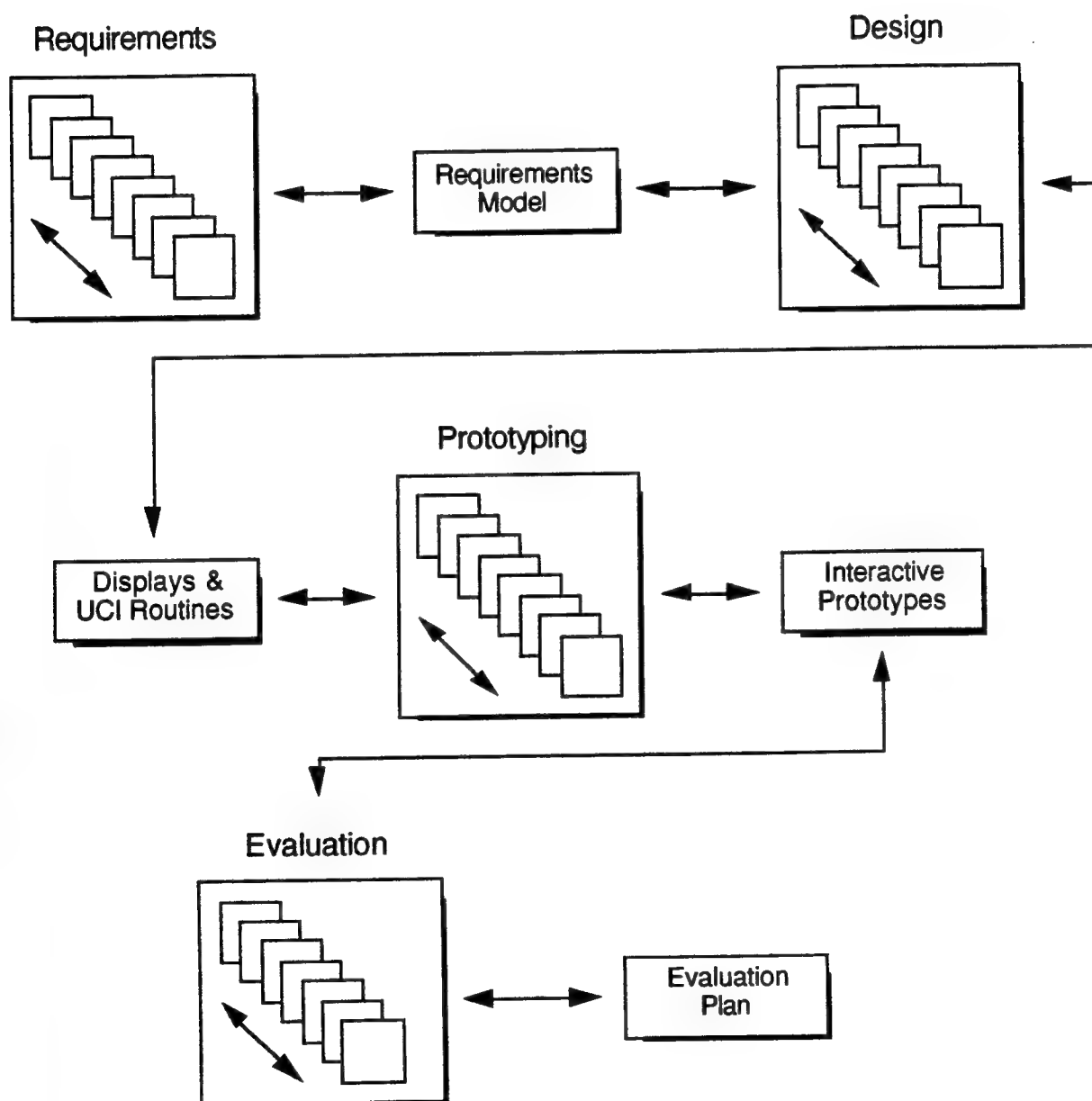


Figure 6: The Design & Prototyping Process

design process in an application that would support the systems engineering of user-computer interfaces and interaction routines -- as suggested above by Figure 6.

The major design process (and workbench) output includes user-computer interface designs, interactive prototypes, and prototype evaluations; these "products" are developed with the assistance of a knowledge base that can accelerate the design ---> prototyping ---> evaluation process.

1.5.2 The FLEX Case Study

The development of the Force-Level Execution (FLEX) prototype at the Air Force's Rome Laboratory (RL) presented an excellent opportunity for applying and evaluating the our methodology for UCI design. The FLEX Program was a collaborative rapid prototyping effort between the Advanced Concepts Branch in the RL/C3 Division and industry contractors. Officers from the US Air Force's major operational commands in the United States, Europe, Pacific and the Far East participated in a review board known as the FLEX Working Group (FWG) to provide an operational end-user perspective. The RL development team took responsibility for developing the user interface.

The Drexel University/George Mason University team worked within this design and prototyping organization to enhance the user-computer interface and interaction process design and prototyping process. This work resulted in a prototype that enhanced performance substantially, and suggested the direction in which implementation should proceed (precisely the goal of the design methodology and the DesignPro system which incarnates the methodology).

The design methodology was employed to define the problem, identify and model the (cognitive) task requirements, integrate the requirements into the System/Segment Specification (SSS), translate requirements into UCI design goals, create an interactive prototype of the FLEX interface, and develop a plan to evaluate the prototype against the existing FLEX interface.

Section 3.0 of this report provides detailed information about the FLEX case study.

1.5.3 The (Cognitive) Task Analysis Process

A major outcome of the project was the development of an entire methodology for identifying and modeling tasks, tasks that feed the larger user and system

requirements modeling process. The step-by-step methodology emerged from the project, a methodology that was developed into a Handbook for conducting tasks analysis-based requirements analyses and modeling, prototyping and evaluation.

This Handbook appears in Appendix C.

1.5.4 The Knowledge-Based UCI Design, Prototyping & Evaluation Workbench

The knowledge-based system -- DesignPro -- that resulted from the research and development currently runs on Apple Power Macintoshes (and other high end Macintoshes). The system supports user computer interaction designers as they identify user requirements (defined as "tasks"), build interactive prototypes (via the implementation of embedded COTS software), and evaluate the prototype(s) to determine if their features should be implemented in production code.

Section 4.0 of this report describes DesignPro workbench in detail, while Appendix A presents the screens from the system.

2.0 UCI Design, Prototyping & Evaluation

The project assumes -- as discussed above -- that the best insight into requirements that should be satisfied in formal designs and development efforts can be obtained from a structured, repeatable requirements analysis, prototyping, and evaluation process.

2.1 UCI Requirements Analysis

The project assumed, therefore, that effort should be directed to the front part of the front-end of the design and development process: user requirements. The form of those requirements is the task. Task analysis is well established as a requirements notation. It is also consistent with our goals to repeat, model, trade-off and prioritize requirements.

Task-based requirements modeling permits reusability and is consistent with current efforts to use objects to support design and development.

Task-based requirements also permits requirements documentation via descriptions of user tasks that can be assessed and prioritized.

While there are certainly alternatives to task-based requirements modeling, we opted for this flexible approach because the alternatives did not satisfy the design requirements unique to user-computer interface and interaction routine design and development. UCI design is inherently task-oriented. This is because of the linkage to UCI display and interaction features and how they can be traced to specific and groups of user tasks that the system needs to support (to, in turn, support the functions that the define the system and the purpose for which the system exists).

Our use of tasks is intended to "trickle up" to functionality. Our requirements template assumes that functional requirements (along with non-functional and purposeful requirements) will define the specification and design process. Tasks lead to functions which lead to purpose. This is the track that DesignPro follows.

The essence of the design process -- and therefore what DesignPro supports -- is requirements modeling. Specific effort was taken to ensure that the "front-end of the front-end" was solid. While we have obviously made some analytical choices, we believe that the existing task-based requirements modeling process will yield diagnostic requirements data, data that can be fed directly into the design and prototyping processes.

Figure 7 presents the elements that together yield a requirements model (which, in turn, becomes the input to the design and prototyping processes).

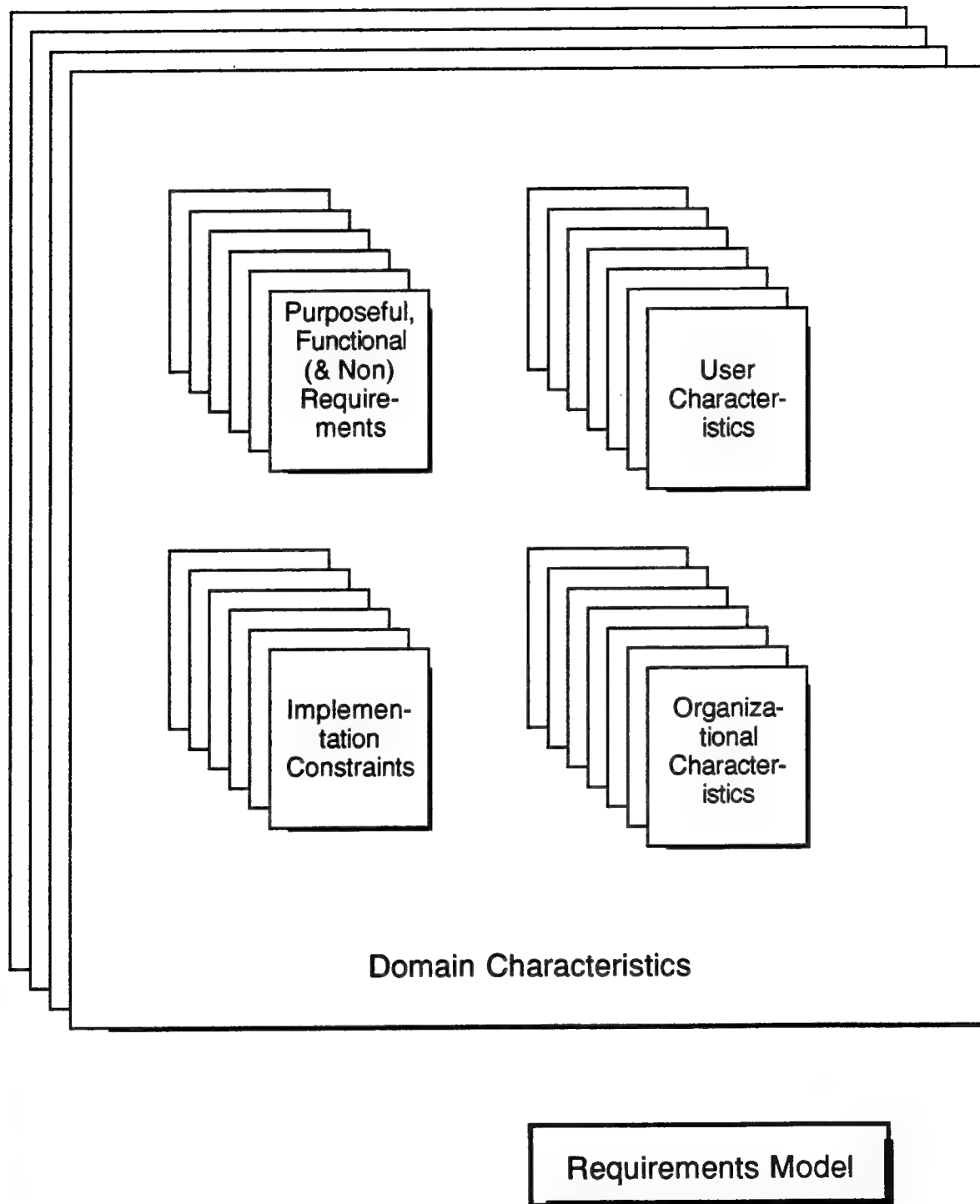


Figure 7: The Requirements Modeling Process

2.2 UCI Features

The DesignPro system provides support for the identification of the UCI design

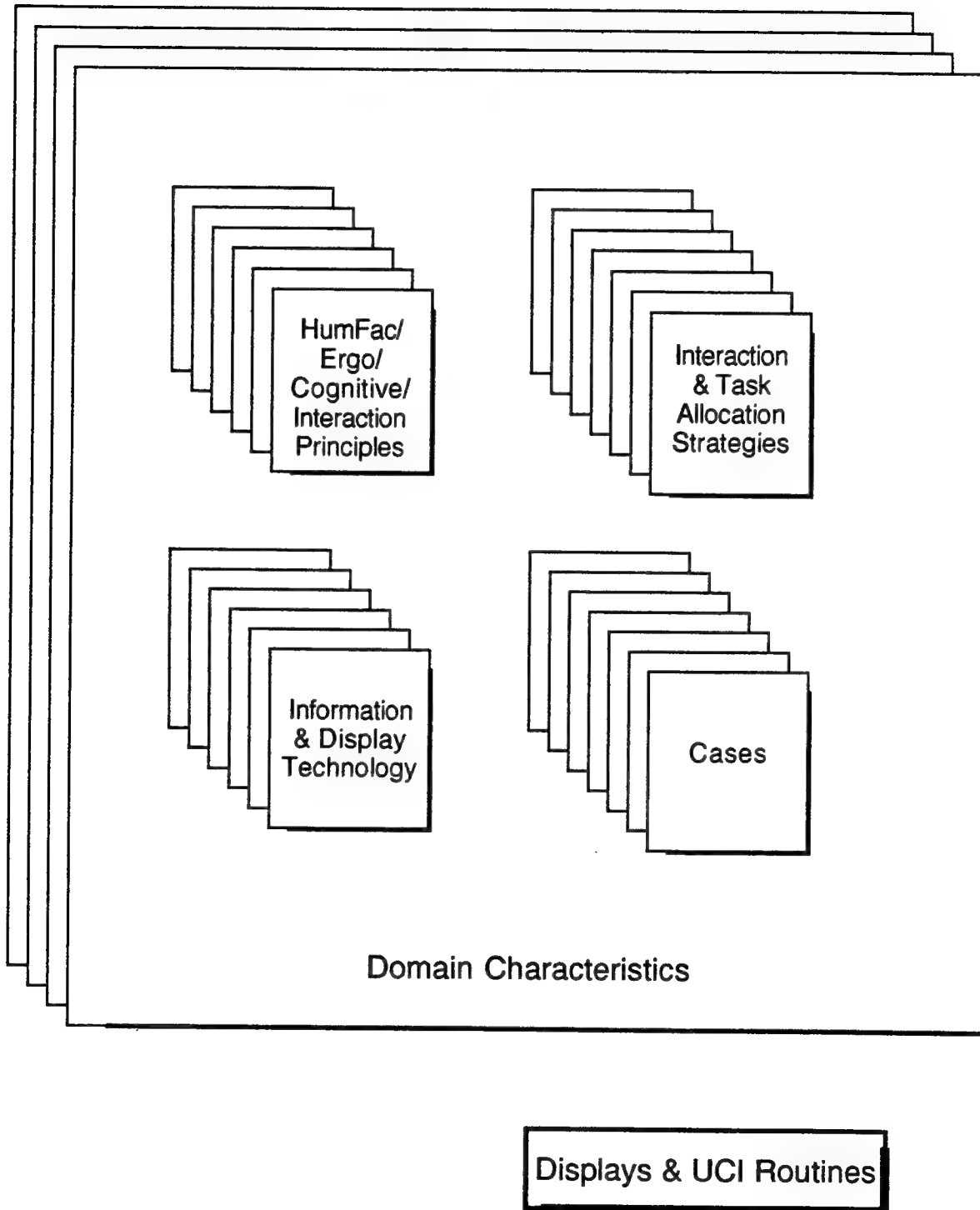


Figure 8: The Display & UCI Routine Identification Process

features that are most likely to satisfy user requirements (given the constraints identified during the requirements modeling phase).

As Figure 8 suggests, there are a number of elements that lead to the recommended displays & UCI routines.

2.3 *UCI Prototyping*

DesignPro supports prototyping.

Prototyping is an absolute prerequisite to writing software requirements specifications. You may think you know the requirements (even for a system enhancement), but the very best you can expect to do is build and test evolutionary prototypes.

The process we advocate (and have embedded in DesignPro) -- and the one that virtually always applies -- is: prioritized requirements (given constraints) ---> throwaway prototyping ---> initial software requirements ---> evolutionary prototyping ---> detailed software requirements specifications.

Detailed software requirements specifications should not -- in our experience -- emerge without prototyping feedback. This is obviously true when throwaway prototypes are built, but also necessary during the initial evolutionary iterations. One can never assume that requirements data is valid; it must be inspected -- and demonstrated -- by lots of eyes, lots of perspectives, and lots of objectivity.

The key lies in "templating" the requirements modeling and prototyping process and in getting experienced professionals to implement and manage the process. The key also lies in self-documenting COTS software that permits group design and communication.

Another key is pragmatism. It's essential that we appreciate the fluid, changing nature of requirements, and the role that prototyping plays in the requirements discovery process.

DesignPro supports the iterative prototyping process via the application of COTS software (which is embedded in the system).

The architecture of the system permits the addition (or deletion) of COTS software as it becomes available or as new tools are identified as "preferred." However, it must be remembered that not all COTS tools are equal, and that some are better suited to do some features prototyping and some less so.

Figure 9 presents the prototyping “template.”

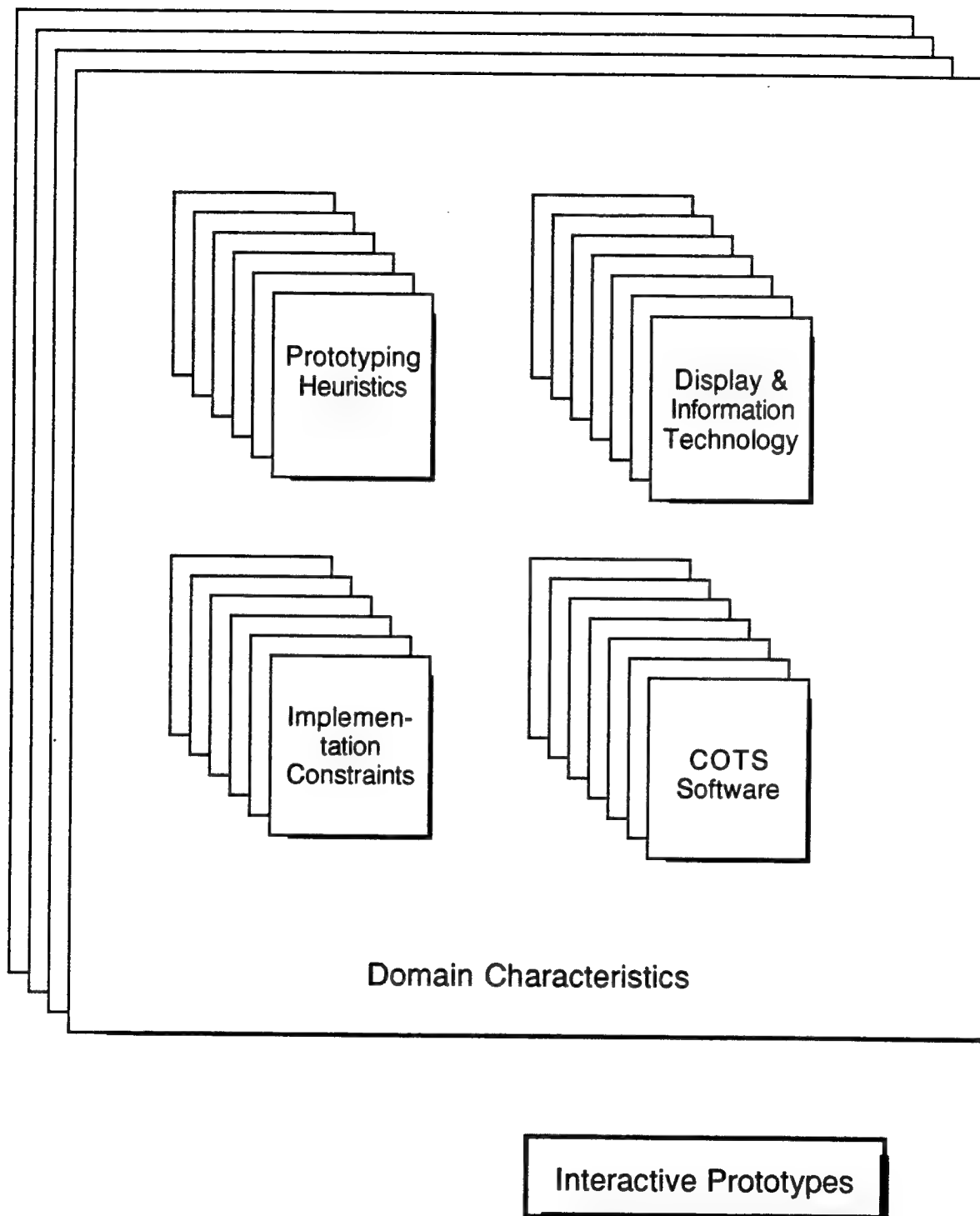


Figure 9: The Prototyping Process

2.4 UCI Evaluation

As Figure 10 suggests, the evaluation process is also straightforward.

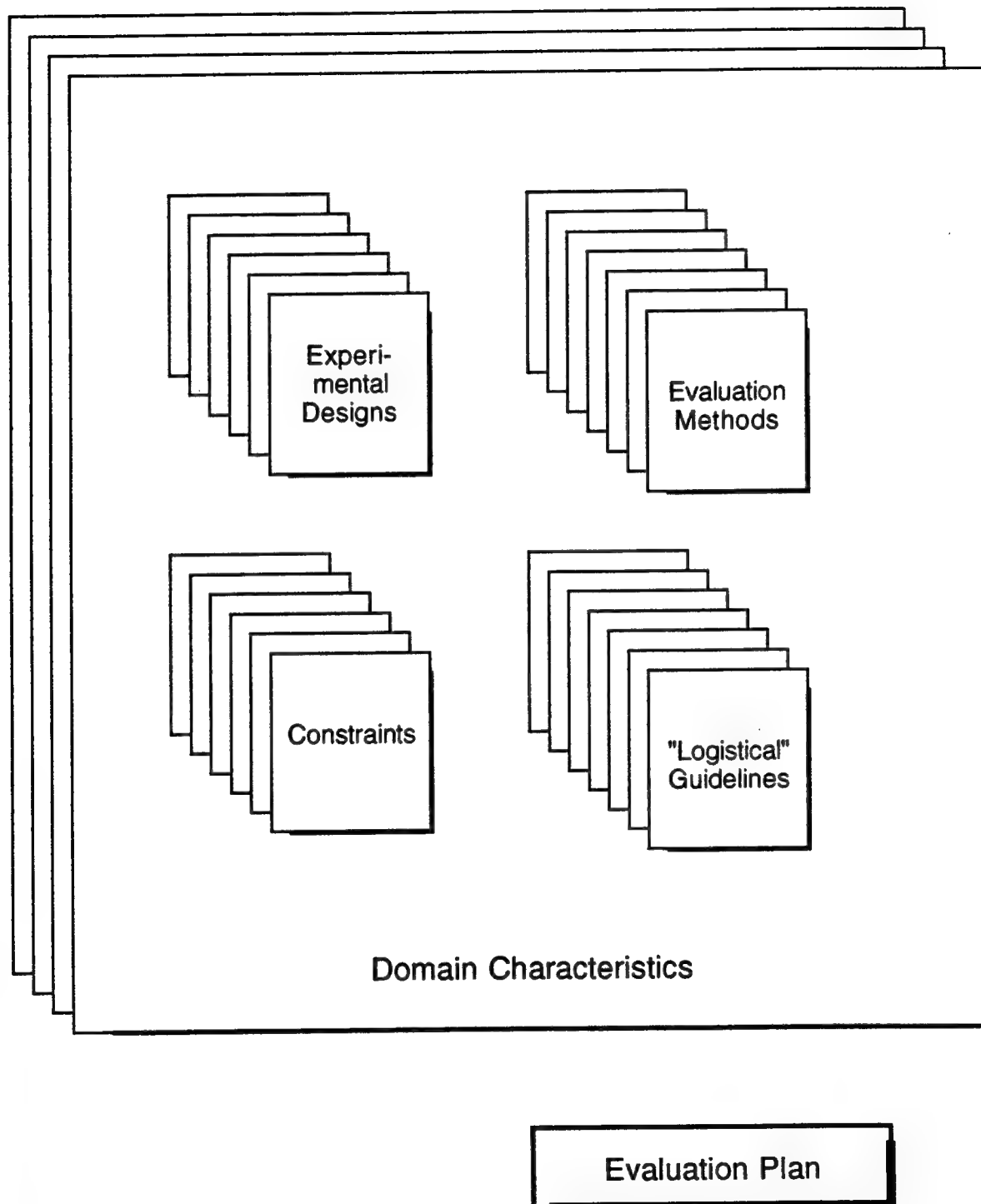


Figure 10: The Evaluation Process

2.5 Knowledge-Based UCI Design, Prototyping & Evaluation

The knowledge base lies at the heart of the system. The approach taken to the identification of design processes is anchored in the generic “objects-attributes-values” approach to knowledge representation.

Figure 11 presents the approach graphically.

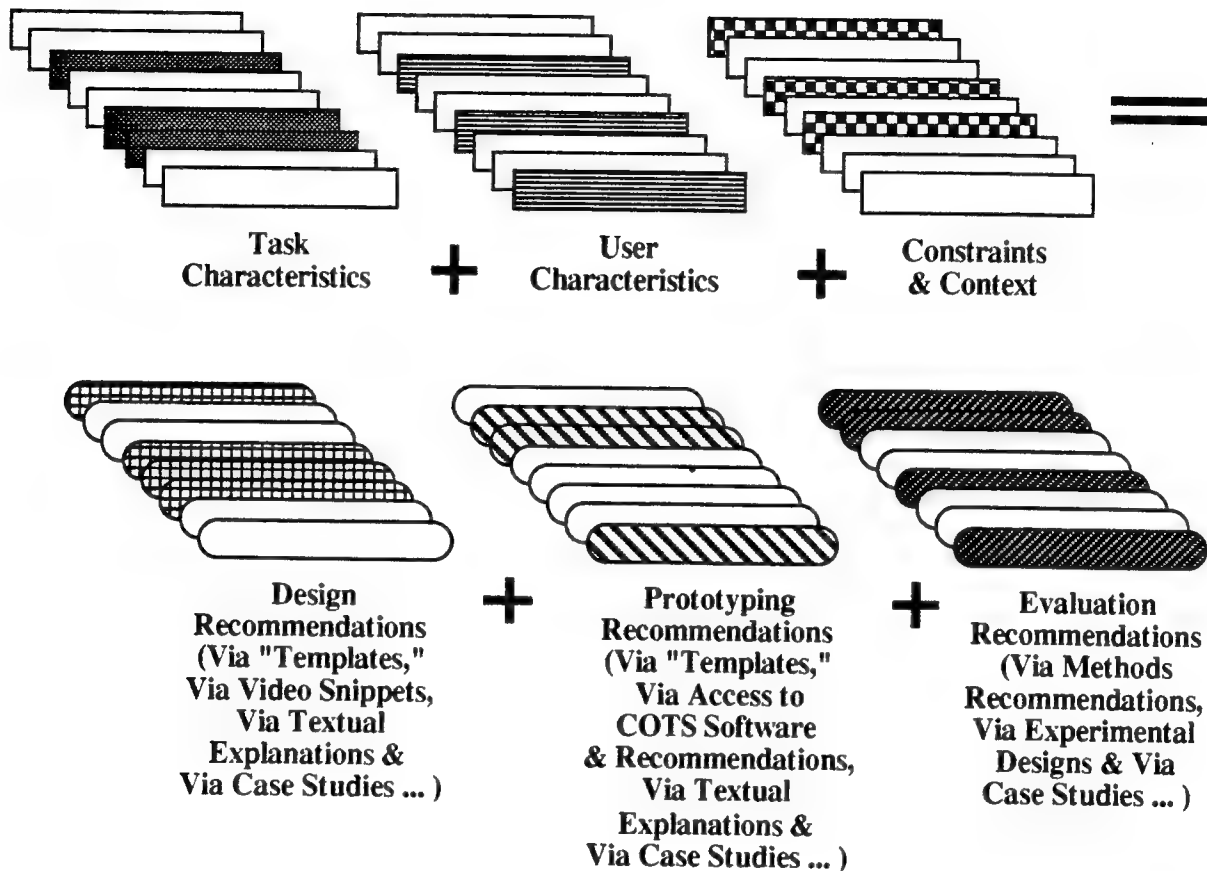


Figure 11: The Knowledge Base Structure

The knowledge base itself was developed from several sources: the case study literature, the experience of the design team, and the unique aspects of the domain that “interpret” the relationships among the objects, attributes and values.

The design and prototyping recommendations DesignPro generates can be traced to these relationships and to the characteristics of the domain that we focused upon: command & control.

3.0 The FLEX Case: Feature Enhancements Via Prototyping

3.1 Defining the Problem

The first step in problem involved defining the FLEX system and placing it in context within the organization. Most of the initial definition was based upon an early version of the FLEX System/Segment Specification (SSS)¹ and the trip reports written by the RL development team after their visits to air operations centers in the United States, Europe, the Pacific and the Far East. Additional information was drawn from the demonstration of the first FLEX prototype for the FWG members.

As indicated in Figure 12, FLEX is part of a suite of systems that supports the Combat Operations Division (COD) of the Air Operations Center (AOC) in the planning and execution of the air missions. FLEX receives the mission plan from the Combat Plans Division in the form of an Air Tasking Order (ATO). Formatted in machine-readable text, an ATO for 24 hours of combat missions may run into hundreds of pages. While the text form permits rapid transmission to the operational units, the ATO is unwieldy and does not provide a sense of the actual mission flows (Figure 13). To compensate, planners and operational staff officers manually create a variety of charts and maps to display mission data in a form that permits multiple data views. The Advanced Planning System (APS) allows the planning team to work with the details of the ATO data using tabular displays of the relevant data bases and automating many of the calculation and charting operations. Since the planners and operational decision-makers use much of the same data and knowledge bases, APS and FLEX share many common screen layouts.¹ The common windows help promote consistency across these two closely-coupled systems.

The combat air operations decision environment is complex and dynamic, involving a high degree of uncertainty combined with time pressure and high threat. The duty officers (DOs) in the COD monitor the execution of the ATO missions and re-plan as required to meet changes in goals and/or available resources.² The various air missions are so interdependent that changes in the availability of a support mission can result in the cancellation or re-scheduling of attack and support missions across the entire ATO. This "ripple effect" makes timely re-planning extremely difficult.

¹ The SSS is the standard document for system-level requirements documentation under the Dept. of Defense software development standard, DOD-STD-2167a.

² The Duty Officer in AOC is a decision-maker, thus, in discussions of FLEX, the term decision-maker (DM) is used interchangeably with duty officer (DO).

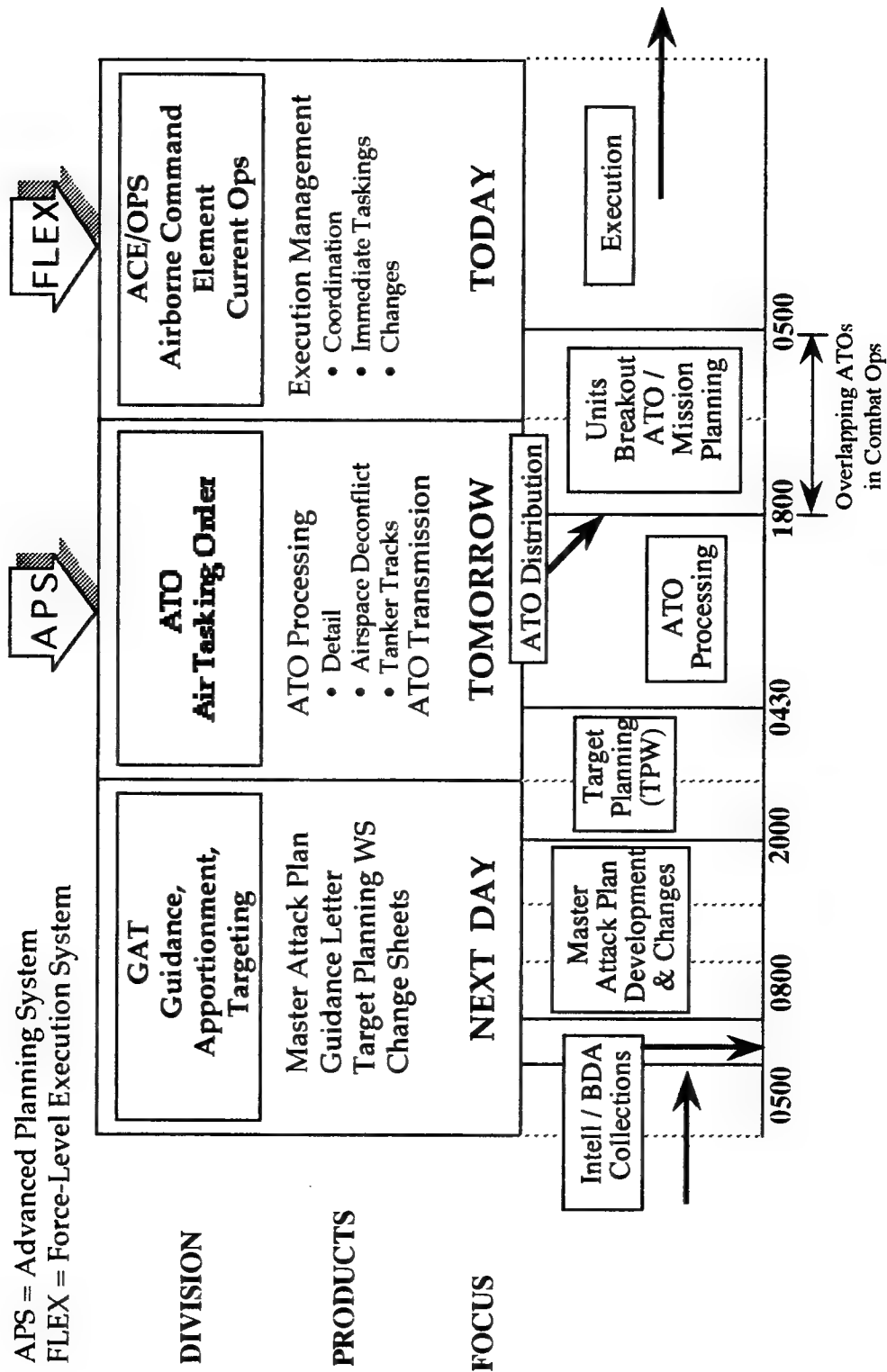


Figure 12: FLEX & the ATO Timeline

```

EXER/FLEX CASE STUDY//
MSGID/ATOCONF/TACC//
PERID/150600Z/TO:160559Z//
AIRTASK/UNIT TASKING//
TASKUNIT/25ARS/OBBI//
MSNDAT/11/-/LINKAGE00/1KC135R/AR/ /BOM/ /126/23600/33600//
AMPN/SHELL//
MSNLOC/151130Z/SHELL/ALT:170/-/2811N04650E//
7REFUEL
/MSNNO/ACSIGN          /NOTPAC   /OFF /ARCT   /TNKR
/FUEL/CMNT
/39      /NOSEGAY       05/2A10A   /12   /   /   /JP4/PRE
/39      /NOSEGAY       05/2A10A   /25   /   /   /JP4/MID
LATE
/62      /NOSEGAY       11/2A10A   /12   /   /   /JP4/PRE
/62      /NOSEGAY       11/2A10A   /12   /   /   /JP4/MID
LATE
//
...
//
TASKUNIT/353TFS/OEPA//
MSNDAT/46/-/NOSEBAG01/1A10A/XCAS/-/B5/-/124/23301/33301//
MSNLOC/150800Z/151000Z/HANDEL/ALT:200/-/2840N05535E//
REFUEL/ROMAN00/21/ESSO/ALT:170/-/10//
REFUEL/ROMAN00/21/ESSO/ALT:170/-/11//
REFUEL/ROMAN00/21/ESSO/ALT:170/ /2//

```

Figure 13: Examples from an Unclassified ATO

The first models developed for problem definition decomposed the monitoring and control, planning, and communications tasks performed by the COD decision-makers in terms of their representation in the first FLEX prototype and the trip reports. These models were iteratively refined as the requirements were identified.

To provide a tractable example, the case study focused only on the FLEX re-planning support to the Tanker Duty Officer (TDO). The TDO is responsible for providing air refueling (AR) support to all scheduled missions that require refueling. Re-planning is required when new missions are created, existing missions re-routed, or air refueling resources change. The TDO performs re-planning tasks as indicated by their own assessment of the evolving situation and as tasked by other duty officers. Although these models were roughed out during

the problem definition phase, most of the detail was developed as part of the in-depth analysis conducted during the requirements identification phase.

3.2 Identifying and Modeling the Tanker Duty Officer's Cognitive Task Requirements

The case study was external to the actual FLEX development effort; therefore, the task requirements identification process began with the examination of system requirements information gathered from a variety of sources including:

- **Document Reviews** - Rome Laboratory (RL) development team trip reports, FLEX statement of work, contract developer's system/segment specification (SSS) and system software design documents, written change requests, and a variety of Air Force manuals and support materials on air refueling operations were reviewed.
- **Interviews** - interviews were conducted with RL team, the contract development teams, FLEX working group (operational personnel from major commands), and tanker operations personnel from Griffiss AFB's 509th Air Refueling Squadron.
- **Observation** - observations were made of FLEX Working Group (FWG) officers interacting with early prototype versions of the FLEX interface.

The user, organization, task, and environmental/situational models evolved into a set of cognitive task requirements (CTRs) that became the design objectives for the interface prototype. These materials were used to iteratively refine the models of the air refueling domain, the TDO and the tanker re-planning tasks.

The remainder of this section reports the requirements gathered from documentation, interviews and observation. A number of graphic hierarchies and taxonomic models were created as the requirements evolved. These were used to develop an understanding of the procedures and information required to accomplish the re-planning tasks.

3.2.1 Defining the FLEX Environmental/Situational Context

It was possible to characterize the FLEX environmental/situational context in terms of its inherent structure, determinacy, boundedness, and complexity. In combat situations, decision-makers in the COD must cope with an environment

that ranges from severely stochastic (e.g., the coordination of a complex array of friendly assets) to indeterminate (e.g., mission perturbations caused by an intelligent adversary). There is a high degree of variability in all the ATO plan components. For example, the decision variables are generally representative, but differ substantially in reliability due to timeliness of updates or their inherent ambiguities. Due to factors such as uncontrollable environmental conditions and the existence of intelligent adversaries, it not possible to completely control the outcomes by manipulating the initial conditions. Thus, the re-planning environment tends to be open and ranges from semi-structured to unstructured due to the high volume of information and potential for "unknown unknowns". Most of the information load under routine conditions is tractable for a well-trained and highly-motivated TDO. Under combat conditions (e.g., 2000 sortie ATO) the tasks become intractable, with information loads exceeding human ability to absorb and manipulate.

Situational complexity ranges from moderately high to very high depending upon the nature and size of the operation. Air refueling is a pervasive support activity and tanker missions are the "tent pole" in air operations. Moreover, tanker operations involve a secondary network of dependencies. The fuel a tanker has available for refueling (i.e., taskable fuel) is dependent upon actual fuel offloads that, in turn, are dependent upon the specific receiving aircraft and the nature of the missions involved. An inability to meet refueling requirements will result in cancellation of missions (direct dependency) with a ripple effect upon the missions which the canceled missions support (indirect dependencies). Due to these extended dependencies, the situational picture becomes less reliable as multiple changes to the ATO are effected during combat execution. As a result, the question is not *whether* the ATO will unravel, it is *how much, in what ways, and when* it will unravel.

This complex situational context has several impacts. First, in response to the domain, the organization must develop the means to make most efficient use of resources in a succession of varying short-term situations. Moreover, the decision-makers must be able to rapidly and effectively exploit opportunities and retain maximum flexibility and adaptiveness in novel situations. There is a potential for misallocation of resources due to the latency between recognition of the situation and internal readjustment. The adaptive strategies required (e.g., rapid re-tasking) may be difficult to coordinate and control due to complex missions interdependencies. Furthermore, achieving the required flexibility may negatively impact the ability to exercise control. Finally, organizational learning may be impaired by the lack of repeated experiences.

To meet the organizational response goals and potential errors, the decision-makers must be provided with information to help them understand the structure of the domain and current problem. For example system-level (i.e., tanker

operations) overview displays can relate functional relationships and provide externalized mental models of the operational domain. Decision-makers also need the ability to adaptively filter information at the required abstraction level, while retaining rapid access to detailed information.

3.2.2 Profiling the FLEX Organizational/Doctrinal Context

The COD is part of a hierarchical organization which has both a vertically and horizontally complex chain of command with a moderately-high interdependency between functional units. The vertical complexity shifts to very high in joint and combined operations that require extensive coordination. The control structures in adaptive decision-making organizations shift in response to changes in the decision requirements. Thus, the general tendency toward the more formal organization evidenced during routine operations shifts during crisis situations to accommodate the requirement for a more flexible response.

During routine operations, the situational context is determinant to moderately stochastic. The threat is low and the environment is relatively static with longer decision horizons. As a result, operations tend to be tightly controlled and decision-making is more formal. Responses to re-planning situations follow more rigid procedures based on specific guidance; therefore, the TDO is less likely to exercise a high degree of personal initiative. Control is communication-dependent and the communication delays between levels of the hierarchy lengthen the time between decision and action. Routine operations afford little opportunity to develop a range of adaptive responses as the TDO never has to push the system to the limit. As a result, during non-crisis operations, the TDO may be ill-prepared for a sudden shift in the environment to a combat state.

In contrast, during crisis operations the situational context is severely stochastic to indeterminate. The adversarial threat of destruction and mission failure is high and decision time is greatly constrained. To facilitate rapid, adaptive responses, operational control is loosened such that the informal problem-solving structures within the COD may dominate the formal structures. As the COD workload increases, the TDO will exercise more individual initiative. Although this provides the TDO an opportunity to extend his/her repertoire of response options, the subsequent relaxation of control may result in local satisficing (that is, solving the sub-unit problem at the cost of larger goals). Intra-COD communication greatly increases and the central role of tanker operations results in a barrage of task alerts to the TDO. Communication delays may impair information gathering and decision implementation required for more adaptive responses.

3.2.3 Profiling the Tanker Duty Officer (TDO)

The profile of the Tanker Duty Officer (TDO) incorporates not only their knowledge of the specific functional tasks assigned to them and their ability to operate the system, but also their understanding of goals and characteristics of the larger domain in which those tasks are performed. The TDO is typically an Air Force major or lieutenant colonel with a moderately high knowledge of the air operations domain acquired through experience, training, and service schools.

Many of the errors in situation assessment may be traced to the DM's knowledge of the operational context. The TDO will have situational models of the domain mostly gained through instruction and exercises and should recognize most prototypical situations. In many cases, for example, the TDO may have wing-level, but not force-level mental models. TDOs without operational experience at the wing or force level will not generally possess wholistic domain models. In addition, although domain-knowledgeable TDOs may exhibit the ability to intuitively interpret novel situations, they may not be consistent in their combination of situational cues. TDOs will generally structure goals based upon learned procedures, direct guidance, and situational models of domain and task. The extent of his/her domain understanding may limit the TDO's ability to resolve conflicts between situational models. Situations triggering multiple models may be interpreted based on the model that is more available or vivid in memory. Finally, the TDO may fail to recognize the degree of uncertainty in current information or the impacts of aggregated uncertainties on the viability of the plan.

The TDOs' knowledge of the specific functional tasks assigned them in the COD may also vary depending upon their previous experiences in combat operations (force and wing level) and training (schools and exercises). TDOs will typically exhibit high ability to perform routine procedures and moderate to moderately-high adaptability under increased workload and novel situations. Their moderate to high task experience potentially triggers errors associated with the heuristics used to reduce the high workloads during ATO execution (Table B-8). For example, in high information volume situations, moderately knowledgeable TDOs may not have adequate schema to distinguish relevant versus irrelevant information. They may also erroneously focus on task features that match stored (especially readily available) schema. Fixation on task features that match well-known (or vividly remembered) situations may prevent the TDO from correctly diagnosing the situation. Furthermore, misdiagnosis may result in the misapplication of a learned response. More experienced TDOs are still vulnerable to a general insensitivity to the potential aggregation of error in the microdecisions performed in multi-stage decision-making. For example, they may tend toward overconfidence in their current decisions and fail to revise their

assessments and decisions when the situation changes. Finally, there is a general tendency for the TDO to think in serial, linear sequences rather than parallel networks of contributing causes and branching consequences of actions that make up the current situation and affect the success of the plan.

The TDOs' system interaction/operation knowledge will typically be the most variable dimension. In the absence of a protracted war, the majority of the officers assigned to the COD will be casual to competent system users. That is, they will not routinely have to operate the system under the time-critical, high workload conditions which characterize combat operations. Adequate operation of the system during routine or training operations will deteriorate under stress resulting in a variety of errors and an increased level of frustration and confusion. Casual system users tend to forget training without use and make mistakes (errors due to wrong intentions) and slips (errors due to unintentional actions). Casual system users rarely remember the system shortcuts that speed up performance of learned procedures and the increased workload will result in greatly impaired performance for all but simplest tasks. The competent user will be able to adapt well-understood processes to increased workload, but still have difficulty with the increase in novel situations. More competent users make mistakes by misapplying learned procedures.

TDOs with less system experience may be confused by their system operation errors. For example, TDOs may make modal errors due to a misunderstanding about current system state. A modal error involves the incorrect use of an interaction procedure that would be correct in another system state. In addition, users may "get lost" in the system, finding themselves in unfamiliar windows or locked out while the system performs an unintended procedure.

3.2.4 Profiling the TDO's Functional Tasks

The TDO functional tasks were reviewed, filtering them through the user, organization, and situational context profiles described above. This process identified several key dimensions which defined task performance and error modes, including:

- Task complexity and difficulty
- Task performance precision and accuracy requirements
- Input and feedback uncertainty
- Task workload and potential stress dimensions

It should be noted that probing task dimensions often triggers further refinement of the other profiles and all of this investigation involved repeated iteration in

both top-down and bottom-up analyses. Figure 14 presents one of the conceptual maps created to describe the response requirements associated with real-time threats that “pop up” during ATO execution.

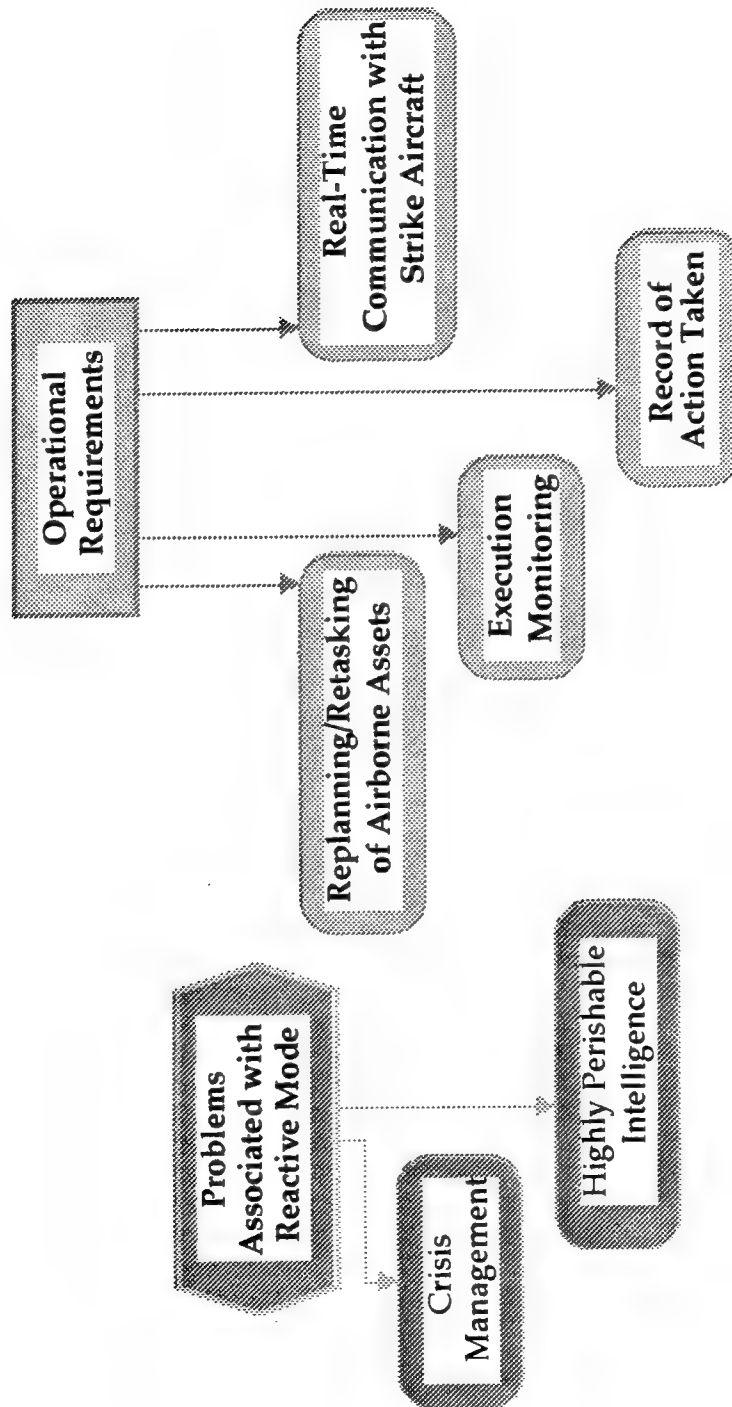


Figure 14: Response to Real-Time (“pop-Up”) Threats

3.2.4.1 Task Output

The TDOs' discrete output unit is the response to a task request for air refueling (AR) support. In a larger sense, the task output is also the overall status of the air refueling plan or the tanker operations system. The TDO is required to respond to a high volume of AR task requests as rapidly as possible; thus, they tend to be extremely intolerant of slow system response or highly complex routines for relatively simple tasks. Air refueling plans have multiple components and TDOs need system supports to prevent their losing track of all relevant plan components. For example, decision-makers need the ability to move through various levels of detail and system supports for structuring the various components to aid in analysis.

3.2.4.2 Task Response

The TDO's response goals are to meet the air refueling requirements of the ATO and maintain a viable air refueling plan for as long as possible. Both the short-term execution goals and overall mission completion goals are very difficult to attain. The system should be designed to offload the TDO of as much of the workload as possible (e.g., by allocation of table look-up and computational tasks to machine). Some of the subtasks (e.g., keeping track of taskable fuel) require high precision that is best allocated to the machine component. For example, the detailed data required for response precision can be maintained and manipulated by machine. In addition, automated updates relieve the TDO from being overwhelmed by the detail.

TDO response frequency during the execution of a major combat ATO is very high. As a result, AR tasks and changes to tanker operations pile up and must be prioritized to ensure the most important are handled as rapidly as possible. Delays in feedback (external or internal to COD) may impair the TDO's timely response.

3.2.4.3 Procedures & Sub-Tasks

AR tasks arrive as discrete messages, but may have to be handled by considering the planning implications of several changes simultaneously. Handling a single AR task involves several steps, including the possibility of activating a ground alert tanker mission or creating a new tanker mission to resolve major changes to the AR plan. In addition, the TDO may have the current working task interrupted by a higher priority task. The requirement for the TDO to simulta-

neously handle the current AR tasks using FLEX while remaining a part of the off-line COD activity (e.g., incoming messages from other sources, conversations with other duty officers, etc.) also contributes to the time pressure experienced. The system must support the TDO's maintenance of situational awareness and task continuity, and complement the team activities of the COD.

AR subtasks are moderately dependent in terms of temporal order (either due to system or procedural constraints) and logical relationships; however, the subtasks are highly dependent with respect to the total AR plan. The overall dependency of AR plan is such that the complexity of relationships exceeds the TDO's ability to handle without support. The TDO needs a way to "step back" from the current situation to see the AR plan as a whole and understand the various direct and indirect dependencies. AR tasks' procedural complexity is moderately high to very high due to the number of subtasks potentially involved and the dependencies between them. Certain subtasks require strict adherence to set procedures; other subtasks may be handled in so many ways that a strict procedure is not prescribed. Where strict adherence to procedures is required, the system support must be designed to constrain TDO from ignoring critical procedures and make those constraints visible to the TDO. In contrast, where flexibility is allowed, the system should facilitate the TDO's ability to manipulate the options and make the affordances visible.

3.2.4.4 Task Input

Many of the input variables in the AR task are moderately predictable due the consistency of operational procedures, basic situational stability, etc. Some input values vary widely in predictability due to inaccuracy of supporting data or novelty of the situation. As a result, the TDO may need to be reminded of the less predictable aspects of the task to ensure that proper attention has been paid to the immediate contingencies ("what-ifs"). For example, variations which follow known patterns under certain conditions may be stored as templates to support faster recognition.

AR tasks are triggered in a very irregular fashion; the TDO generally cannot predict the flow of AR tasks with other than very gross metrics. The TDO cannot control the occurrence of the stimulus (AR task), but can control the order of response among tasks of the same priority. Although alarms may be shut off and incoming AR tasks acknowledged and set aside for later response, an AR task remains an open issue until changed by the TDO's response. Thus, the TDO may need to regularly review open requests and reorder priority under heavier workloads.

3.2.4.5 Task Feedback

More than 50% of the AR subtasks involve decisions based on feedback from previous responses. As suggested above, the TDO must respond to some high priority AR tasks immediately, while other tasks may be postponed temporarily. For this reason, the TDO needs to know when tasks will become critical to help in prioritizing numerous tasks with the same priority. Feedback to the TDO from other COD duty officers on actions taken is immediate; however, feedback from the tankers and other flying missions may be delayed by hours. As a result, feedback reference may be ambiguous as actions taken early in ATO day may be superseded by later events before feedback reaches the TDO.

As the ATO day progresses, TDO plan refinements may be entirely dependent upon the projected effects of plan changes for which there has been only partial feedback. The required reaction time for decisions is much less than the typical feedback lag and the TDO may have to make many dependent decisions long before feedback on one decision is received. This can result in over- or under-adjustments to the AR plan. To compensate, the TDO needs a means to model potential effects of actions against a likely model of the current situation. The secondary effects of feedback lag impact the effectiveness of the decision-maker's learning and experience. False assumptions due to feedback lag can generate inaccurate mental models regarding cause and effect relationships. For this reason, the TDO needs support for trying (and retracting) optional courses of action before committing to decisions.

3.2.5 Profiling the TDO's Decision-Making Tasks

The general characteristics of the FLEX functional tasks apply to all the duty officer positions. For this reason, most of the functional task identification described above was accomplished before the case study was narrowed to tanker re-planning operations. As the requirements identification shifted to the detailed profiles of the decision-making tasks, the focus narrowed to the Tanker Duty Officer (TDO) with particular emphasis on the decision-making activities involved in re-planning during ATO execution. Figure 15 presents one of the conceptual models developed to help identify the key activities and variables in tanker re-planning tasks. This section presents the decision-making requirements identified and modeled. The TDO's cognitive task requirements are considered in terms of:

- *Stimulus* - situational input
- *Hypothesis* - situation interpretation
- *Option* - course of action review and selection
- *Response* - coordination and execution of chosen option

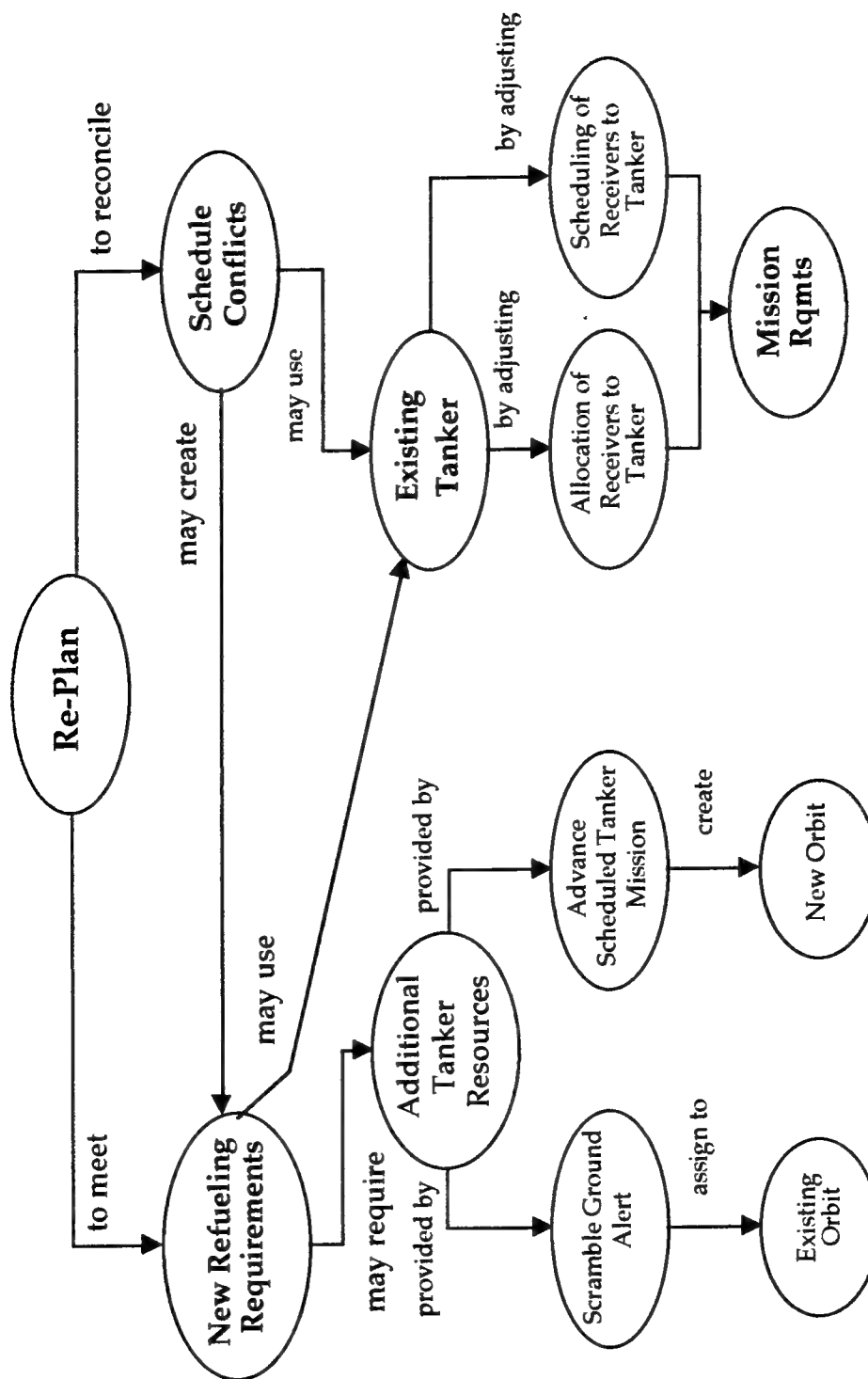


Figure 15: Replanning Tasks

3.2.5.1 Stimulus - Characteristics of the Situational Context and Data Inputs

Situation monitoring for the Tanker Duty Officer (TDO) in the Combat Operations Division (COD) is largely reactive. Unlike real-time tactical monitoring, the TDO is not directly manipulating the environment on a minute-to-minute basis. Instead, monitoring and decision-making are carried out in a time-constrained environment, primarily driven by incoming update alerts or task requests. Because important operations information may exist on multiple screens, the TDO needs to have changes brought to his attention. Pop-up display of new task requests makes detection of discrete air refueling (AR) requests automatic; however, the TDO may have considerable difficulty detecting underlying trends in tanker operations due to variations in the timeliness of updates to key variables.

Tanker operations information is primarily quantitative; qualitative information is inferred through maps and mission flows. The TDO's situational awareness requires supports for tailoring displays to filter, sort, and organize information. In combat situations, the volume of incoming updates to tanker operations data exceeds the human's ability to absorb or manipulate within the time requirements. The FLEX system automates the detailed updates and alerts the TDO to conflicts spawned by changes in resource availability.

FLEX information on tanker operations exists primarily as detailed data tables with summary information available in the Tanker Status Display Board. The Map Graphic window charts information such as the locations of bases, tanker orbits and tracks, routes of planned missions, and defensive coverage. FLEX users can filter the information presented to suit the requirements of their decision tasks. The Marquee is a graphic interface to much of the FLEX database. The Marquee's adaptable display presents some of the operational dependencies across the ATO timeline through a database feature that allows the user to sort and "bundle" dependent missions. However, the FLEX filtering does not adequately reduce workload due to complexity and information volume. Due to the screen layouts (particularly in the Tanker Worksheet), the TDO is still required to do some mental computation and make notes to keep track of certain variables. The TDO needs system support to reduce off-line mental computation and other memory requirements.

Tanker operations decision variables (e.g., fuel requirements, etc.) are generally understood and representative. When the required data are current, the variables are reliable for calculation and decision-making; however, this is not always possible due to communication failures or other feedback delays. Furthermore, the TDO may not fully assess the impacts of situation and options based on

displayable information; there are potential "unknown unknowns" in combat operations which undermine the representativeness and reliability of standard decision variables. Mis-perception of the situation due to incomplete or ambiguous information can lead to any or all of the following:

- Focus on irrelevant information
- Selection and/or fixation on an incorrect explanation or solution
- Incorrect interpretation of cues
- Insensitivity to missing information

Given these potential cognitive failures, the TDO may benefit from displays of system models or goal states to aid in:

- Identifying problems
- Defining causal relationships;
- Identifying missing information;
- Interpreting ambiguous cues; and
- Reducing over-confidence in decisions based on uncertain information.

The existing FLEX interface addresses some, but not all of these needs.

3.2.5.2 Hypothesis - Situation Assessment Task Characteristics

Several factors combine to make hypothesizing for situation assessment difficult. Although the TDO is familiar with all the activities of tanker operations, there is situational novelty inherent in the ways the variables may combine in combat. Joint service and multi-national (combined) operations add extra layers of complexity and novelty to tanker operations. Finally, the unpredictability of an intelligent adversary may result in an unfamiliar sequence of events. The combination of novelty with the crush of information flow may distract the TDO from seeing the underlying similarity to more familiar situations. To relieve the TDO, certain routine aspects of AR re-planning may be allocated to machine processes.

Situation assessment for air refueling operations is semi-bounded with a moderate number of hypothetical possibilities to explain current AR plan status; however, the number of hypotheses may seem greater under heavy workload situations. The TDO needs relief from complex detail through aggregated displays and interaction with models that help to identify the differences between the current and goal states. Goal-oriented displays of tanker operations also help to maintain focus on critical variables and serve as templates for analogies to familiar

situations. Finally, to understand the potential direct and indirect effects of the current situation, the TDO needs a means of viewing the consequences of actions across the ATO day.

TDO performs situation assessment tasks in a time-critical, quasi-real time environment. This requires prioritizing backlogged tasks and often means trading off time to fully analyze situation in order to process more AR tasks in a shorter period of time. Comments for the FLEX Working Group (FWG) after all three prototype reviews indicated that the visual momentum involved in using FLEX was still relatively low due to the requirement to use operational data scattered across several windows to accomplish any task. To relieve the time pressure in situation assessment, the TDO needs "at-a-glance" displays that do not require hunting or elaborate manipulation of detail to get to the relevant information quickly. In addition, the TDO should not be burdened with off-line computation.

Most of the inferencing required for AR replanning is within set bounds, involving well-known parameters; however, the complexities of multiple receivers and their dependent missions creates a hidden network of inferences with varying degrees of certainty. This multi-dimensional network of inferences is very memory-intensive. To compensate, the TDO must use workload reducing heuristics that may introduce bias errors. The TDO needs displays which support inferencing based on accepted operational procedures. In addition, supports for option exploration should reduce the number of inferences and relieve the workload on TDO by portraying the current (and projected) state to compare with immediate and longer-term consequences across the network of tanker operation dependencies.

3.2.5.3 Option - Course of Action Decision Tasks

The number of possible options to a given air refueling (AR) situation are semi-bounded (as to the limits of available resources, etc.), but sufficient in number that the TDO faced with a large number of outstanding AR tasks is often overwhelmed by the resulting plan complexity. In addition, AR mission goals may shift several times in a relatively short period of time, requiring a re-evaluation of priorities, updates, and recalculation of projected changes in AR plans. Most of the conflicts and effects are predictable, but the number of conflicts spawned in interdependent missions by even a small plan change make manual manipulation intractable. Furthermore, the uncertainties and inherent complexity make outcome values for changing AR plans difficult to project despite the TDOs understanding of the fundamental variables.

The TDO needs facility to quickly package responses for less complex, more routine changes. The TDO needs some means of rapidly understanding the fundamental effects of an option under consideration. Ideally, the system display should support the decision-maker's rapid mental simulation to accept or reject the option as feasible. Although evaluating AR re-planning options is manually intractable under high workload situations, the problem is sufficiently bounded to allow for machine support in several areas, including:

- Rapid recalculation of all dependent mission data to compare options
- Mapping of restructured dependencies
- Highlighting any resulting conflicts

To filter out the best option configurations, the TDO needs tools that allow rapid scoring of options against basic criteria with pre-determined or adjustable weighting. Where rankings are similar, the TDO needs displays that model or simulate the projected consequences for a given option to compare with other relatively equivalent options. Finally, the TDO needs to be able to step back from detail and view AR operations in terms of higher level goals. For example, predictable goal changes may be combined into contingency scenario templates and displayed to the TDO as advance notice or incorporated into a rule-based advisor.

Outcome uncertainty for most AR plan components is moderate, but predictable. Nevertheless, the broader the scope of the plan change, the less certain the outcome. TDO choices at time t may leave them more or less vulnerable at time $t + 3$. The potential vulnerability to later requirements changes (i.e., contingencies) is even more uncertain and difficult to factor into the decision. Combined levels of uncertainty add to the intractability of option evaluation. Moreover, feedback may not be timely, goals may change several times, and there is a very high penalty for making poor choices. The current FLEX system does not reflect the uncertainties aggregated into projected outcomes of AR plans. The system's ranking of options treats all quantitative data as being 100% certain. Thus, it is possible to have two equally ranked options, yet be unaware of their highly disparate levels of certainty. The TDO needs supports for understanding the degree of uncertainty inherent in a particular option.

3.2.5.4 Response - Planning, Coordination and Execution of Decisions

Air refueling plans are operational hypotheses involving multiple assumptions and inferences about the current situation and the causal relationships that predict outcomes. AR execution in high sortie ATOs can make use of pre-planned contingencies (e.g., by activating orbits and routes, launching ground alert tanker

missions, selecting alternate recovery bases, etc.) to handle many of the plan changes. Extensive re-planning is required when major changes are made during execution (i.e., the addition of a large, high-priority mission; multiple failures; or resource losses). Re-planning decisions are further complicated by the difficulty of tracing all possible consequences of actions taken. The TDO needs support for decomposing new goals into AR subtasks and means-end restructuring of AR plans to meet new requirements.

Execution in tanker operations requires coordination with other DOs in the COD, with airborne forward control units, the affected strike wings and support operations. During joint and combined operations coordination also involves other services and national forces. AR coordination must take place within the decision horizon and is affected by the organizational shifts that occur in crisis conditions. Communication requirements for coordination (i.e., management of message traffic) impose processing loads on the system which constrain the design options. Reformatting to meet messaging standards qualitatively changes information passed and may affect its interpretation at the receiving end. Although coordination is handled through SODO and ATO distribution chain, the TDO needs support for understanding the potential coordination ramifications of options related to interdependencies and communication delays.

Execution of AR plan changes is a highly dependent, multi-phased control process. Multiple phases increase coordination requirements and can affect the feasibility of certain options due to the limits of the decision horizon. Delayed feedback may be incorrectly associated with the wrong phase and cause the TDO to over-correct. To track execution, the TDO might benefit from a display of goals and subgoals with current execution status.

3.2.6 The FLEX Cognitive Task Requirements (CTRs)

Appendix C presents a summary of the issues raised during the CTR identification phase for the FLEX Case Study. The goal of the requirements identification process was to re-examine the available requirements definition resources and enhance the existing FLEX requirements specification. Thus, many of the functional requirements identified are represented to some extent in the FLEX System/Segment Specification (SSS) and the FLEX prototypes. These high-level functional requirements for the UCI design group under three main support requirements: performance improvement, distributed decision-making, enhancing the decision-maker's knowledge base. Table 1 breaks these requirements down into their respective components.

Support for Improved Performance

- Support rapidly adaptive response.
- Provide DM most accurate, relevant information and technological means to combine and interpret information.
- Offload DM of as much of the workload as possible.
- Support pattern-matching, analogical reasoning, and other means for improving assessment in novel situations.

Support for Distributed Decision-Making

- System must support the TDO's maintenance of situational awareness and task continuity, and complement the team activities of the COD.
- Provide means to maintain overall control to meet mission objectives without direct review of every micro-decision by senior command
- Optimize for fast communication to improve coordination and minimize authorization delays.

Support for Development of Decision-Making Knowledge

- Make use of natural or domain knowledge in the interaction symbology to allow the user to interact with the task in the most familiar terms.
- Display structural information (i.e., functional cause and effect relationships) to aid development of mental models and support wider knowledge of response options.
- Provide doctrinal/procedural overview displays to support interpretation of and effective response to novel or rare events.
- Provide varying levels of explanation to support the construction of more robust mental models.

Table 1: High-Level Functional Requirements for the UCI Design

3.2.7 Specific Cognitive Task Requirements

Appendix C presents a complete list of the cognitive task requirements and related issues raised during the requirements identification phase. It was necessary to narrow the scope of the Tanker Re-Planning Case Study to three key CTRs, unrepresented in the FLEX SSS and unmet in the FLEX Prototype 3. These included requirements to:

- Adjust the problem viewpoint (level of detail)
- Focus attention on the key decision variables
- Compare response options in terms of potential consequences

First, the TDO needed a way to “step back” from the detailed data with an overview of tanker operations. This was, in part, a response to the time horizon of the TDO’s decisions and the varying degrees of timeliness and precision connected with the updates to the database. Small changes to the published ATO which must occur rapidly (e.g., last-minute re-routing of a mission to another tanker for refueling) are handled in the air by forward controllers. The TDO makes decisions involving a somewhat longer decision horizon and needs to work with an aggregated display of the entire ATO day. Second, the TDO needed a display simultaneously presenting all the critical decision factors. The working group participants complained that key information was distributed across several displays, requiring the user to jump around and make notes off-line. Finally, the TDO needed a support for mentally simulating the chain of consequences (e.g., changes in critical values) associated with feasible options. Answering these requirements without sacrificing access to detail became the central goal of the interface re-design.

The complete list of cognitive task requirements presented in Appendix C was integrated into the FLEX System/Segment Specification (SSS). It was also used to distill the design goals for the based UCI prototype.

3.3 Integrating Cognitive Task Requirements into the System Requirements Document

The Department of Defense development standard for software systems specifies the format and content of system-level requirements documented in a system/segment specification (SSS) document. Although the FLEX case study focused on the decision activities of the Tanker Duty Officer, the CTRs had to be identified and represented in the higher level format of the FLEX SSS. This integration involved distilling the findings from the requirements review presented in Appendix C and matching them to the relevant system specifications in the

existing FLEX SSS. In many cases, the FLEX SSS already contained statements which incorporated the content of the CTR. Occasionally, the statements were modified to improve their precision. In addition, items were appended to stated requirements to detail functionality specified by identified CTRs.

For example, feature visibility -- facilities which enable the operator to control the visibility of all feature overlays (i.e., to enable or disable display of feature data) -- include:

Requirements

- a. The operator shall be able to select the visibility of . . .
- b. The operator shall be able to create, store and select preferred feature visibility defaults to filter or highlight missions/features, including:
 - 1. **Specific ATO time range (current or near future operations)**
 - 2. **Missions/features affected by change/update**
 - 3. **Missions/features in conflict (current or projected conflict)**

Figure 16: Example of a CTR Integrated in the FLEX SSS Document (Additional Tasks Requirements Appear in Bold)

3.4 Translating Requirements to an UCI Design Concept

The cognitive task analysis repeatedly raised certain cognitive aiding issues. These cognitive aiding requirements aggregate into categories of design goals representing situational awareness and understanding, attentional focus, reduction of mental workload, problem perspectives, option evaluation, decision control and guidance, interface operation and error control. The last two goals involved requirements that were adequately addressed in the existing FLEX prototype and lay outside the specific interests of this research.

The remaining six belong to the general category of improving decision-making. These requirements were addressed to some degree in the FLEX SSS and the FLEX prototype designs.

Each is re-capped briefly below.

Goal 1: Support for Situational Awareness and Understanding

- Provide display features (e.g., overview screens) to help the user develop mental models of the operational environment.
- Make the sources and extent of uncertainty explicit.
- Provide templates of various known patterns and causal conditions to support faster recognition.

Goal 2: Support for Focus on Goal/Decision-Relevant Information

- Provide goal- or decision-oriented displays to focus attention on relevant information and support
 - Identifying the situation and/or problem;
 - Defining causal relationships;
 - Identifying missing information;
 - Interpreting ambiguous cues; and
 - Reducing over-confidence in decisions based on uncertain information.
- Provide predictable goal changes in contingency scenario template displays.

Goal 3: Support for Understanding of Operational and Domain Dependencies

- Provide system-level (i.e., tanker operations) displays to convey interdependencies and situational overviews.
- Example: the TDO needs ability to display integrated tanker-receiver dependencies, mission flows on all active tanker orbits and fuel available.

Goal 4: Support for Reducing Mental Workload

- Provide system support to reduce off-line mental computation and other memory requirements.
- Provide an option to use supports (e.g., table look-up tasks) and reminders.
- Provide and propagate automated updates to relieve the TDO of the overwhelming task of maintaining detail.

Goal 5: Support for Viewpoint Adjustment

- Provide the TDO the ability to adaptively filter information to permit the required abstraction level, while retaining rapid access to detailed information.
- Provide the ability to “step back” from detail and view AR operations in terms of higher level goals and the various direct and indirect dependencies.
- Provide “at-a-glance” displays that do not require hunting or elaborate manipulation of detail to get to the relevant information quickly.

Goal 6: Support for Option Comparisons

- Provide a means of viewing the consequences of actions (including the indirect effects) across the ATO day.
- Provide support for trying (and retracting) solutions before committing to decisions.
- Provide a means for a rapid mental simulation to accept or reject the option as feasible.
- Provide displays which support inferencing based on accepted operational procedures.
- Provide support for rapid scoring of options against basic criteria with pre-determined or adjustable weighting.
- Provide displays that model or simulate the projected consequences for a given option to compare with other relatively equivalent options.
- Provide support for understanding the degree of uncertainty inherent in a particular option.

In addition to the immediate benefit of improving performance, Goals 1 - 3 have the potential to enhance long-term performance by developing and reinforcing the mental models that produce a more robust decision-maker knowledge base.

The FLEX Tanker Case Study focused on the immediate benefits of performance improvement derived from the six design goals. Figure 17 maps the interdependencies associated with the individual goals. Research indicates that the quality of situation assessment and ability to preview the effects of decisions improves decision performance (Klein *et al*, 1992; Klinger *et al*, 1993; Raphael, 1991). In particular, improving the DM's understanding of the causal dependencies that underlie a situation and the consequences of a given course of action can help to reduce decision error often associated with complex decisions (Cohen *et al*, 1985; Reason, 1990; Senders and Moray, 1991). The keys to situational awareness and understanding lie in the DM's ability to:

- Filter the relevant situational cues from the complex barrage of data
- Combine the cues to make inferences about the situation (Andriole and Adelman, 1989)

Selecting the appropriate level of detail and focusing on decision-relevant information assists the filtering process; while an understanding of the operational and domain dependencies -- the causal networks -- provides a framework for combining information to make inferences. Relieving the DM of certain detailed mental operations (e.g., calculations, table look-up operations, and various memory tasks) and providing mental organizers (e.g., decision-structured displays) permits the focus of mental resources on the critical decision tasks. Finally, the ability to compare options in terms of potential consequences of actions taken is enhanced by the DM's focus and understanding.

The tasks identified for the FLEX Tanker module during the requirements identification phase and incorporated into the six design goals above map to four CSE design principles. These principles, with the associated design goals in parenthesis, include:

- Presenting a system-level model relating the relevant decision variables to focus the decision-maker's attention and guide the selection of appropriate detail to further inform the decision process (Goals 1 - 6);
- Integrating all the key decision factors in one display to eliminate unnecessary jumping from screen to screen (Goals 2 - 5);
- Making the current system (i.e., tanker operations) state visible to highlight the areas requiring correction (Goals 2 - 5);
- Relieving the DM of calculation and memory tasks (Goals 2, 4 and 6);
- Making the consequences of options visible for comparison and evaluation (Goals 1, 2 and 6).

The first two principles were drawn primarily from the ecological interface design research by Jens Rasmussen and his colleagues (Rasmussen and Vicente, 1989; Vicente and Rasmussen, 1992) and represented in guideline form in Rasmussen and Pejtersen (1993) and Rasmussen *et al*, (in press). In addition, research on the design of integrative displays (Bennett *et al*, 1993) provided further insight into the ways decision cues can be combined in symbolic displays whose decision-aiding "emergent" features are only apparent in that combined form. Finally, the tactical decision-making research by MacMillan and Entin (1991) illustrated the decision performance value of unifying the key decision factors in a single window. The three remaining principles reflect guidance that may be found in all standard guideline sources.

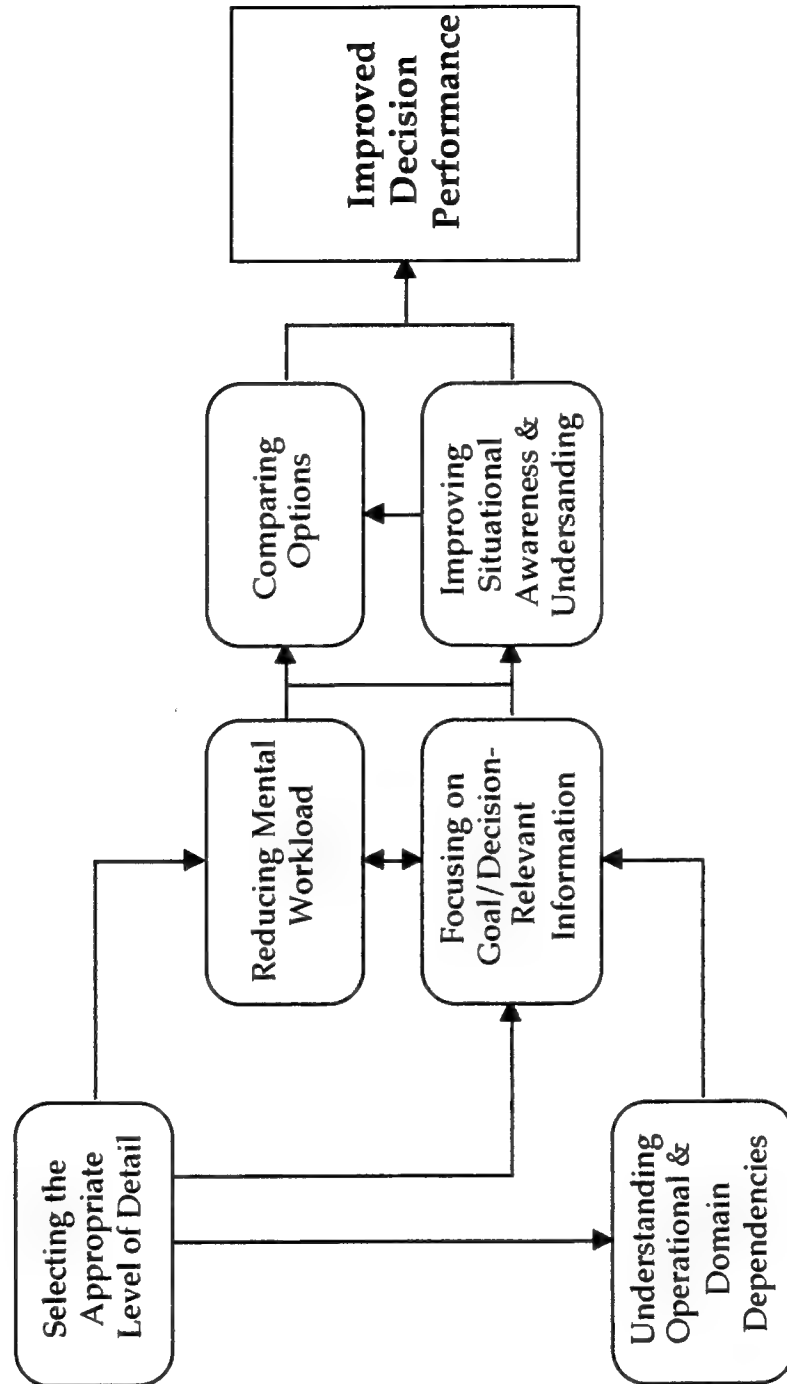


Figure 17: Relationship of FLEX Design Goals to Overall Goal of Improved Decision Performance

The guidance from these principles drove the design of an additional window for the FLEX Tanker DO called *Option View*. (Figure 18). The *Option View* window incorporates a number of UCI responses to the design principles identified. First, the window presents a high-level system model of current tanker operations displaying the active tanker missions at their orbit locations across the 24 hours of the ATO. The receiver contacts are mapped across time against the assigned tanker mission to highlight their flow in terms of density and timing. Conflicts are highlighted in red to draw attention; changes in the tanker or receiver missions are highlighted in yellow. The taskable fuel remaining is displayed above each tanker mission and relieves the DM from having to make the calculation. Second, to facilitate comparison, two options may be compared simultaneously against the planned ATO. (The actual large-screen monitor used for the Air Force FLEX prototype would support comparison of more than two options.) The comparisons present the effects of allocations in terms of changes to the taskable fuel remaining, timing of receiver contacts, and density of assigned receivers against the tanker.

3.5 Developing an Interactive Prototype of the UCI Design Concept

The FLEX ATTD is a technology demonstration program that is intended to evolve into a fielded system. Given the author's external role in the FLEX ATTD, the FLEX Tanker Case Study made use of a throwaway prototype to evaluate the UCI design impacts on decision performance. For evaluation and comparison, both the FLEX tanker module displays and the revised UCI design were implemented in an interactive prototype. The essential features of the existing FLEX windows were mocked-up to allow for rapid prototyping of the key decision factors presented in each window (Appendix G). The extensive searching, sorting and tailoring capabilities of these displays were not represented in order to focus the evaluation on the decision-making tasks rather than the interface manipulation tasks. The evaluation prototype was developed in SuperCard[®] on an Apple Macintosh[®] with a high-resolution RGB color monitor. To facilitate non-intrusive, automated data collection, the software program includes routines to record time-stamped information about the user's interaction with the interface.

3.6 Evaluating the UCI Design Concept

In rapid prototyping development efforts, software evaluation goes on continuously as functional modules are developed and integrated. In similar fashion, UCI concepts and features may be evaluated early in development as

design hypotheses. Such early evaluation is particularly important when the

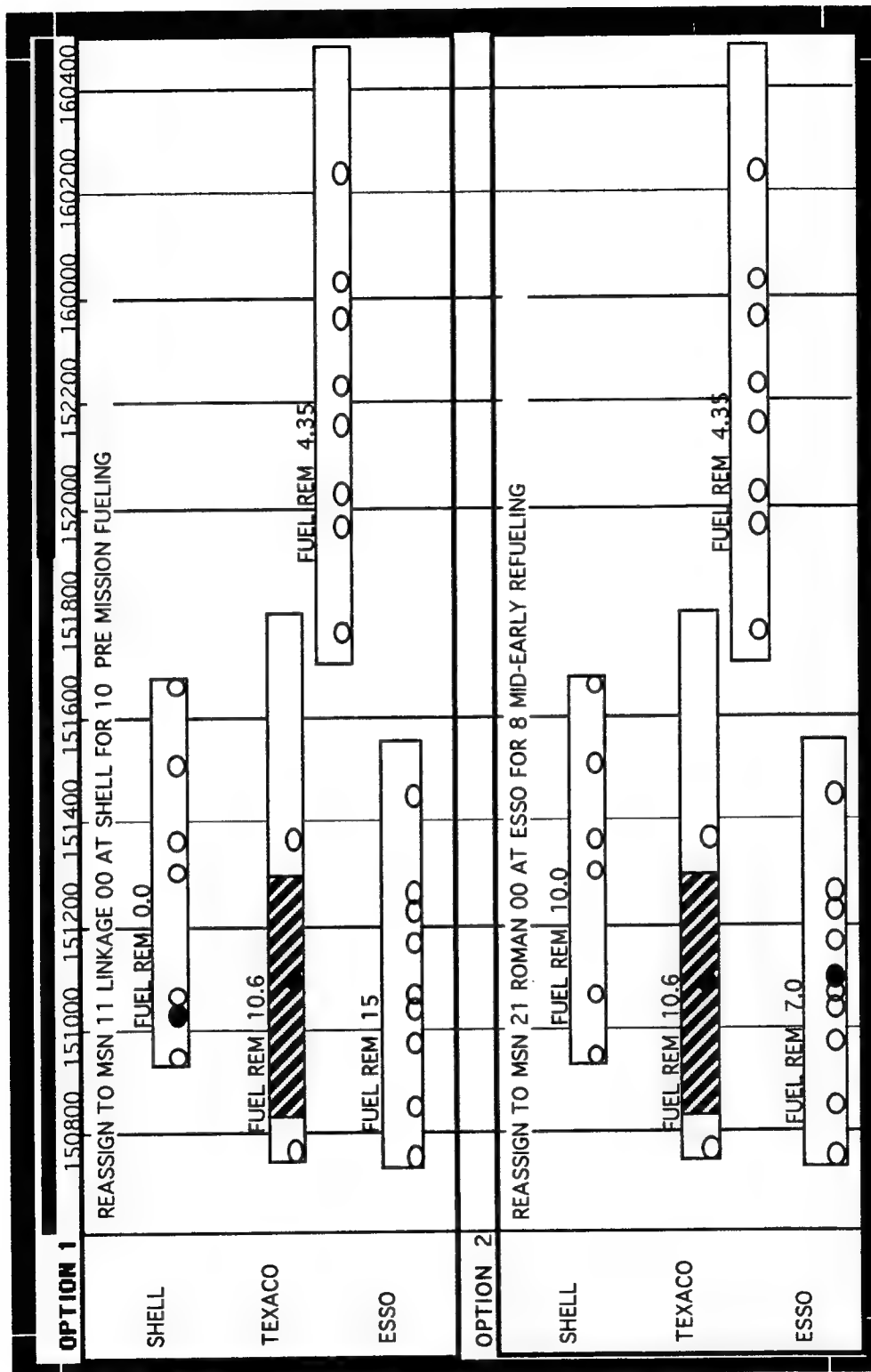


Figure 18: Option View

system contemplated will comprise a major change to the decision-making organization. Early concept evaluations are also useful for evaluating the value-added by incorporating advanced UCI technologies.

In addition, to the narrowly focused evaluations conducted throughout the life-cycle, the overall UCI design must be evaluated as part of a total prototype evaluation. This allows the designers to examine the flow of interaction between the user and the computer and explore interface problems that may not surface in limited studies. Overall evaluation is best conducted using subjects that represent a cross-section of the target end-user population. Although the FLEX Case Study only focused on a small subset of the larger FLEX system, the case study evaluation was conceived in terms of a complete review of the UCI concept in the prototype.

3.6.1 Developing Evaluation Goals

The fundamental hypothesis of the cognitive systems engineering framework is that using the approach should highlight the critical cognitive task requirements and, by guiding the translation of these requirements into design concepts, result in changes in the system which, in turn, result in changes in task performance. The evaluation of the FLEX Tanker Module Prototype sought to validate the approach by demonstrating an improvement in decision performance along three dimensions: situational awareness and understanding, option evaluation, and cognitive workload.

3.6.2 Selecting Evaluation Methods

The evaluation goals identified were very specific to the cognitive task requirements and unique features of the tanker operations domain. For this reason, it was critical to evaluate the task interaction concepts as well as the information presentation aspects of the UCI design. The RL version of the FLEX prototype did not have facilities for setting up multiple small trials. More importantly, the interface was both "fragile" (i.e., prone to frequent crashes) and very difficult to learn. The prototype developed for evaluation focused on the decision tasks and minimized system operation tasks by pre-formatting the highly customizable FLEX windows so that any window called by the user would display its information to best advantage. This was done to eliminate performance variation due to differences in system operation skills. The high level of domain and task knowledge that characterized the target users suggested that subjects for the interaction should be drawn from Air Force officers with a common level of knowledge and experience in tanker operations.

As indicated previously, the framework for the evaluation of the FLEX UCI design was built upon a multi-dimensional view of the factors contributing to effective decision-making performance. The fundamental hypothesis for evaluation may be stated as follows:

UCI designs based upon the approach to identification and specification of cognitive task requirements will result in improved decision-making performance ...

This high-level hypothesis was broken down into measurable factors with respect to three dimensions: situational awareness and understanding, option evaluation and selection, and cognitive workload. Each dimension was represented by one or more design goals that, in turn, were the subject of one or more sub-hypotheses and measures. Figure 19 maps the six evaluation hypotheses and related measures to these three dimensions. Each dimension is discussed in turn below.

Dimension 1: Situational Awareness and Understanding

- **Design Goal:** The presentation of information was designed to highlight and relate key decision factors at the appropriate level of abstraction to relieve DMs from the requirement to accomplish this integration in their heads.

Hypothesis 1.1a: Decision-makers presented an integrated model of the "system" and critical decision variables will more accurately focus their information search than those not supplied with the integrated model display.

Hypothesis 1.1b: In the absence of a fully integrated model display, decision-makers will compensate by selecting the displays which partially integrate key variables.

Measures:

1. Time-stamped Process Trace of Information Views Used
(Comparison with decision model of where critical decision information is located)

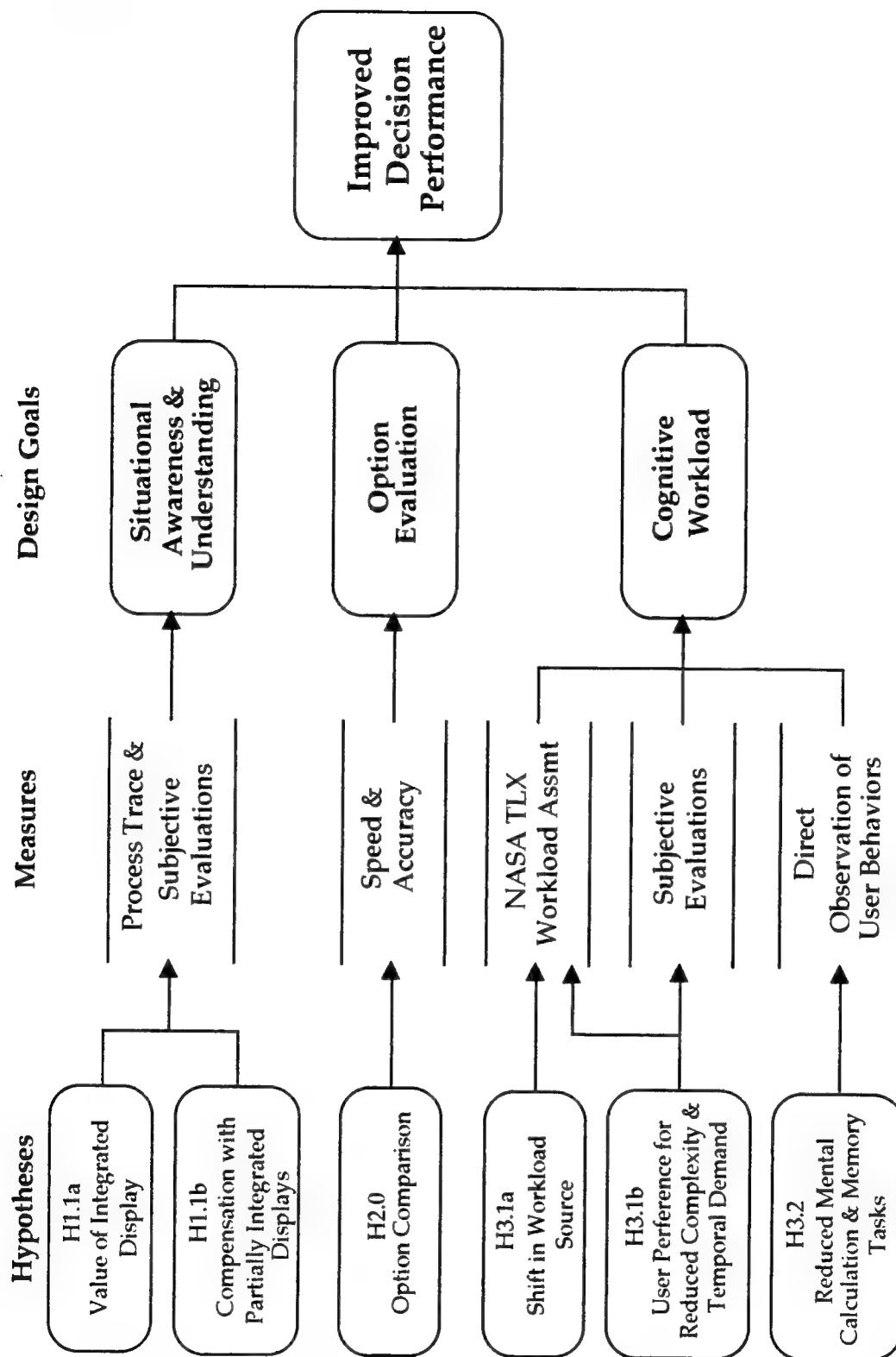


Figure 19: Relationship of FLEX Evaluation Hypotheses & Measures to the UCI Design Goals

- Comparison of mean frequency of window selection
- Process trace (precisely where user went when)
- Comparison of mean duration (seconds) spent viewing each window

2. Subjective Interface Evaluations

(Comparison of interface/task means based upon users rating on discrete scale of specific window's usefulness in four decision tasks)

- | | |
|---------------------------|----------------------|
| •• Problem Identification | •• Option Evaluation |
| •• Situation Assessment | •• Option Selection |

Dimension 2: Option Evaluation

- Design Goal: The information presentation and interaction was designed to allow exploration and comparison of two or more options in terms of their consequences across time.

Hypothesis 2.0: Displaying the changes in the critical variables to allow simultaneous exploration of two or more options will improve option evaluation and selection performance.

Measures:

1. Speed (comparison of mean times to make individual decision - trial and sum - by interface)
2. Accuracy
 - Comparison of mean score on selection of "better" option across trials, users, and interfaces
 - Comparison of ANOVA on scores across trials, users, and interfaces ("better" option determined by previously established experts' model rating options based on taskable fuel remaining and receiver "density" function)

NOTE: Interface exposure order effects were compared to evaluate the potential task and interface learning interaction across sessions.

Dimension 3: Cognitive Workload

- Design Goal: Reduce the users' experience of cognitive workload due to mental demand and time-pressure by designing the information presentation as a "system model" representing and relating critical decision variables.

Hypothesis 3.1a: When other task factors are held constant, the perceived workload associated with time-pressure and problem complexity will be greater for decision-makers working without integrated displays.

Measure: NASA-TLX workload assessment.³

- Comparison of the percentage of total workload attributed to temporal and mental demand depending upon interface used

Hypothesis 3.1b: The subjective evaluation of interfaces will favor those interfaces associated with lower cognitive workload ratings (i.e., those that reduce task complexity in terms of mental and temporal demand).

Measures:

1. NASA-TLX workload assessment

- Mean percentages by interface
- Mean total workload by interface

2. Subjective Interface Evaluations

- Comparison of mean subjective evaluations interface effectiveness across decision tasks (problem identification, situation assessment, option evaluation, option selection)
- Review of open-ended written and verbal impressions of interfaces (audio recording of discussion after final session) vis-à-vis task requirements
- Design Goal: Display the changes in the critical variables to relieve the decision-maker of the extra cognitive workload involved in mentally simulating the comparative effects of the options. Allocate tasks, such as calculation of numerical values (e.g., fuel remaining), to the computer

³ NOTE: NASA TLX is a subjective rating of the user's perception of the source of task workload across multiple dimensions (e.g., mental demand, temporal demand, own performance, frustration, effort, etc.)

as calculation of numerical values (e.g., fuel remaining), to the computer to relieve users of mental calculation.

Hypothesis 3.2: Decision-makers provided integrated displays (i.e., those presenting calculations of all key variables) for comparing the options will not make off-line notes to support their mental simulations.

Measure: Direct observation - collection of session materials for review (i.e., did the users make notes and calculate values while using the interface)

The prototype evaluation was specifically designed to explore the constructs behind the cognitive task requirements and demonstrate the range of information that could be gathered and analyzed quickly. The various measures selected were chosen for their presumed validity as measures of the criteria of interest, but preference was given to methods that were either very quick to analyze or could be automated in the software of the interface prototype. For example, process measures were chosen which could be captured and compiled automatically rather than employing a team of observers, transcribers, and coders to collect and format verbal protocols. The subjects were provided several opportunities to comment on the nature of the interaction and the information presentation. As much as possible, these subjective data were collected in structured formats that facilitated rapid coding and analysis.

Since a sufficiently large group of representative users is difficult to obtain for long periods of time, the evaluation sessions were designed to require each participant to commit to only two half-day interaction sessions. Counter-balanced exposure and a repeated measures design provided sufficient power to achieve significant results with a total of twelve subjects.

The results of the evaluation generally supported all hypotheses. Comments from the subjects after exposure to both interface designs strongly favored the addition of the *Option View* window. Moreover, the subjects' difficulties with the tasks when using the original FLEX interface conformed to the errors predicted during the requirements identification.

3.7 The Cognitive Task Analysis Handbook

The concepts, methods and techniques that guided the design and evaluation of the FLEX interface and prototype lead to the development of a Handbook for repeatable UCI design, prototyping and evaluation. This Handbook appears in Appendix C.

4.0 The UCI Design, Prototyping & Evaluation Workbench

4.1 Hardware/Software Configuration

DesignPro is an Apple Macintosh-based application that was developed in Supercard (by Allegiant Software).

The system also uses a FileMaker Pro data base of "snippets" -- video clips of UCI features stored as QuickTime videos.

DesignPro has embedded COTS software that supports the prototyping process.

The system was configured for a 16" monitor and requires 24K of RAM and 350K of hard disk space to run efficiently.

4.2 Operation

The system is "user-friendly." Those with UCI design, prototyping and evaluation experience will find the interface intuitive; those with relatively little experience will find the system easy to use -- especially via its embedded tutorial (see below).

The following 18 figures illustrate DesignPro's capabilities.

Figure 20 presents the master menu. It identifies three primary activities areas -- requirements modeling, prototyping and evaluation -- and a secondary area -- UCI sampling, which houses the video snippets of selected UCI features and COTS prototyping software.

Each icon represents a functional area available to the designer. The icons represent a top-to-bottom sequential process, though the designer is not bound to proceed sequentially.

The "UCI Sampling" activity area can be accessed at any time.

Figure 21 presents the domain menu. The only domain in the system is "command and control." Other domains can be added to the system as knowledge is modeled in the other areas.

The icons at the bottom of the screen in Figure 21 indicate the navigational capabilities of DesignPro.

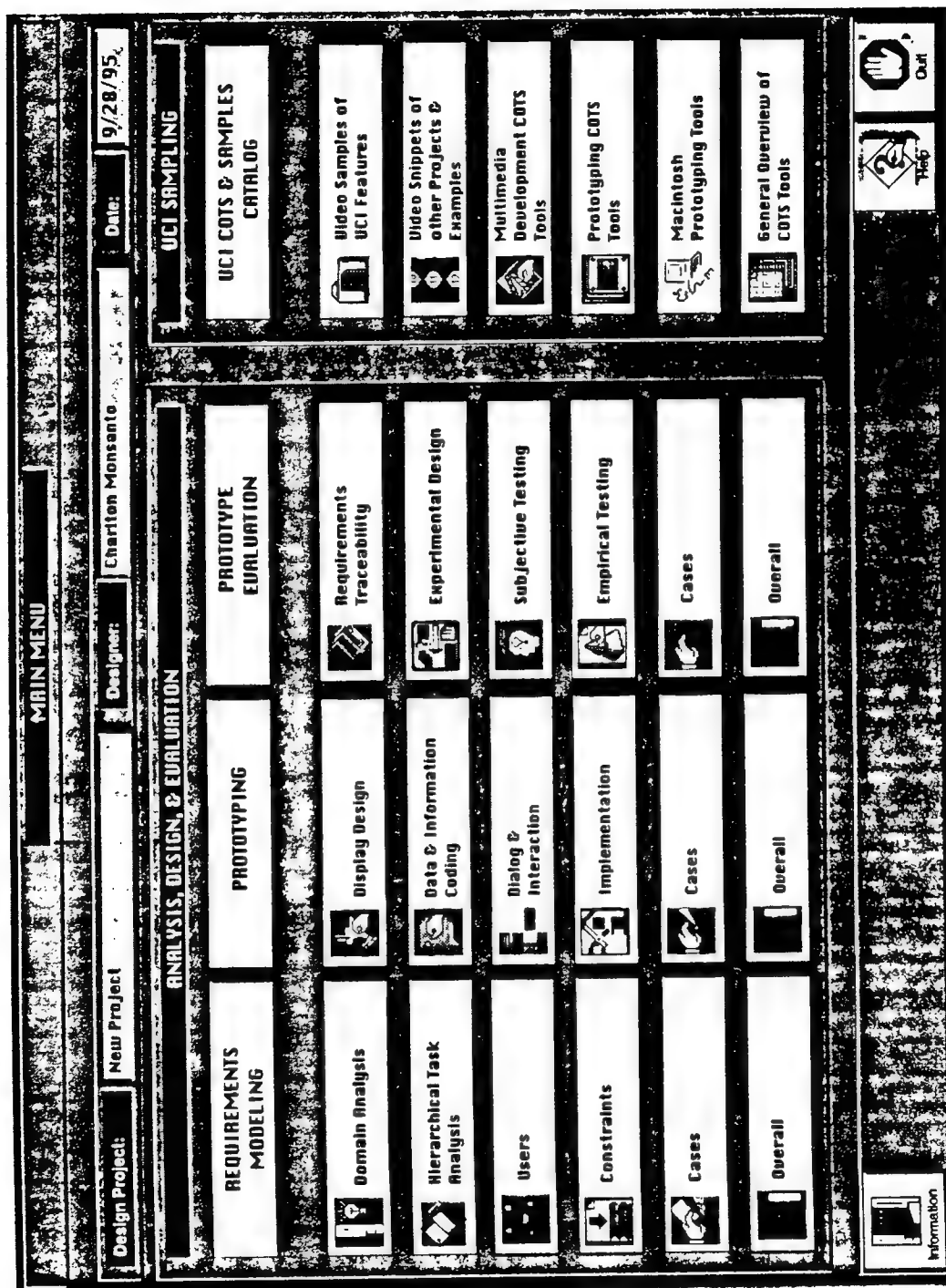


Figure 20: The Master Menu Structure

AVAILABLE DOMAIN GUIDANCE

PLEASE SELECT DOMAIN(S) FROM THE LISTING ABOVE BY CLICKING ON THE NAME

SELECTED

DOMAIN

<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	

DESCRIPTION FOR:

The domain of military command & control refers to actions and reactions that military organizations take as they attempt to achieve missions with objectives. The domain includes activities that can be described along the "sense" <--> "think" <--> "act"



Information



Main Menu



Overall



Help



Quit

Figure 21: The Domain Selector

Figure 22 takes the user to the hierarchical task analysis capability of DesignPro. This capability permits designers to identify the tasks that the user of the interface will have to perform.

There are several levels available to the designer as well as a horizontal capability that permits 5 top level tasks, 20 second level tasks, and 60 lower level tasks.

The "task map" at the top of the screen permits designers to travel across the task hierarchy.

The hierarchical task analysis capability also permits each task to be annotated.

Figure 23 indicates how the system asks designers to characterize each task. A series of judgments must be rendered to permit the tasks to be profiled. "High," "medium," and "low" ratings are permitted.

The system also permits each judgment to be annotated.

Figure 24 requires the designer to identify the constraints that will bind the design, prototyping and evaluation process.

These constraints address the following:

- Target Platform
- Display Device
- Display Device Size
- Operating System
- Management
- Hard Disk Storage
- Graphical User Interface
- RAM

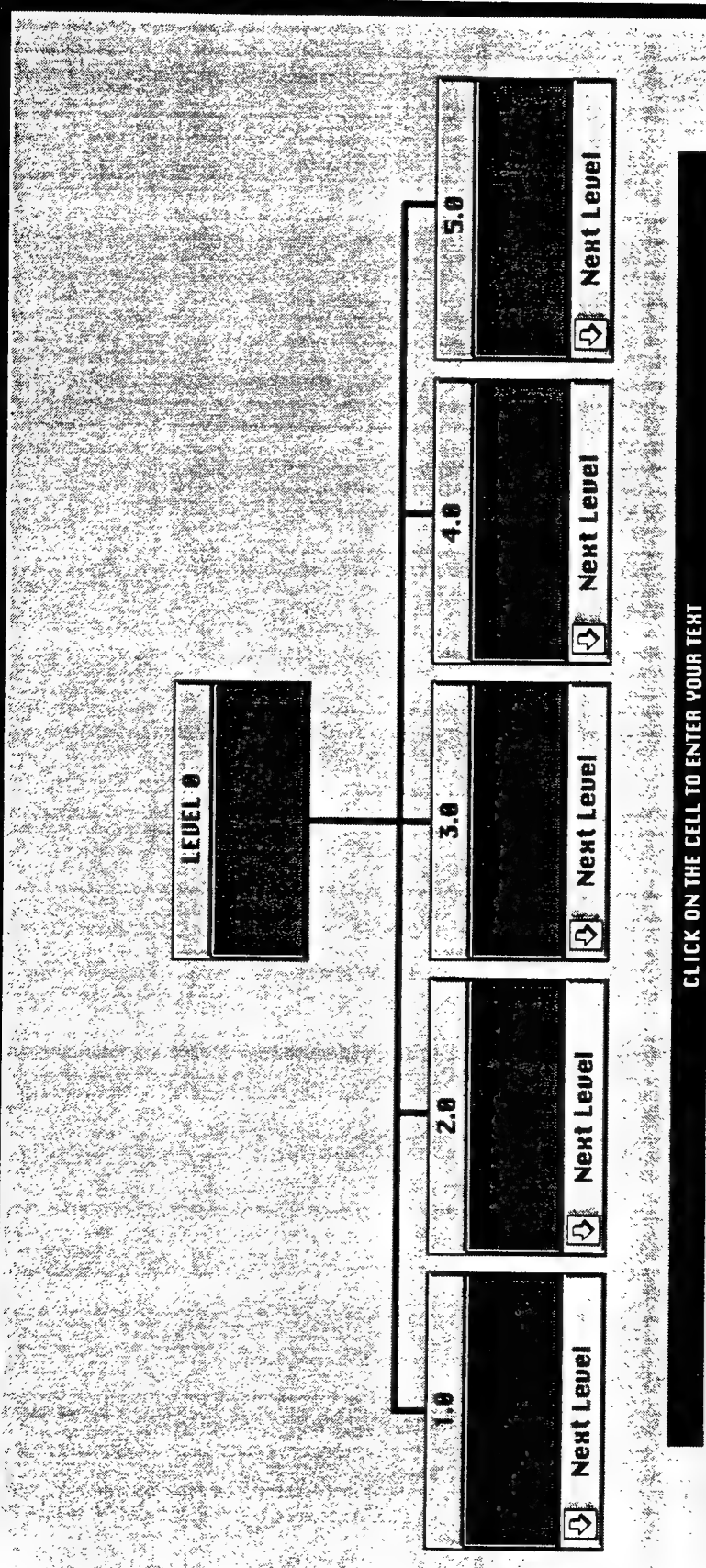
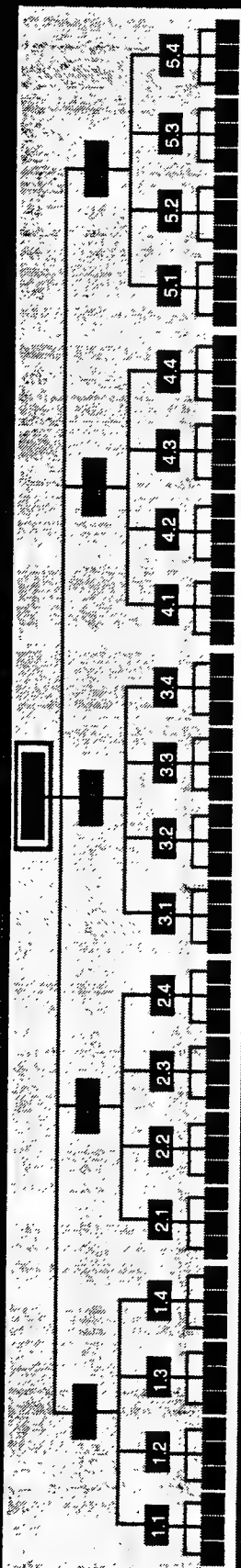
Data about these constraints help determine what is possible.

Figure 25 provides the user with some "cases" with which to compare the design problem at hand.

"Case-based reasoning" is a popular analytical technique that permits problem-solvers to better understand a problem at hand via the inspection of past problem-solving.

Several cases are embedded into the system; the system can receive any number of cases.

HIERARCHICAL TASK ANALYSIS



Information
 Main Menu
 Overall
 Display Design
 Data & Information Coding
 Dialog & Interaction
 Help
 Quit

Figure 22: Hierarchical Task Analysis

Figure 23: Task Profiling

CONSTRAINTS			
Target Platform <div>IBM Compatible</div> <div>Macintosh</div> <div>Workstation</div> <div>Other</div>	Operating System <div>IBM Compatible</div> <div>Macintosh</div> <div>Workstation</div>	Graphical User Interface <div>IBM Compatible</div> <div>Macintosh</div> <div>Workstation</div>	RAM (MB) <div></div> <div></div> <div></div>
Display Resolution <div></div>	Implementation Team <div></div> <div></div>	Management <div>Schedule</div>	Hard Disk Storage (MB) <div></div> <div></div> <div></div>
Display Buffer Size <div></div> <div></div> <div></div>	Budget <div></div>		

Information

Main Menu

Overall

Help

Quit

Figure 24: Assessment of Constraints

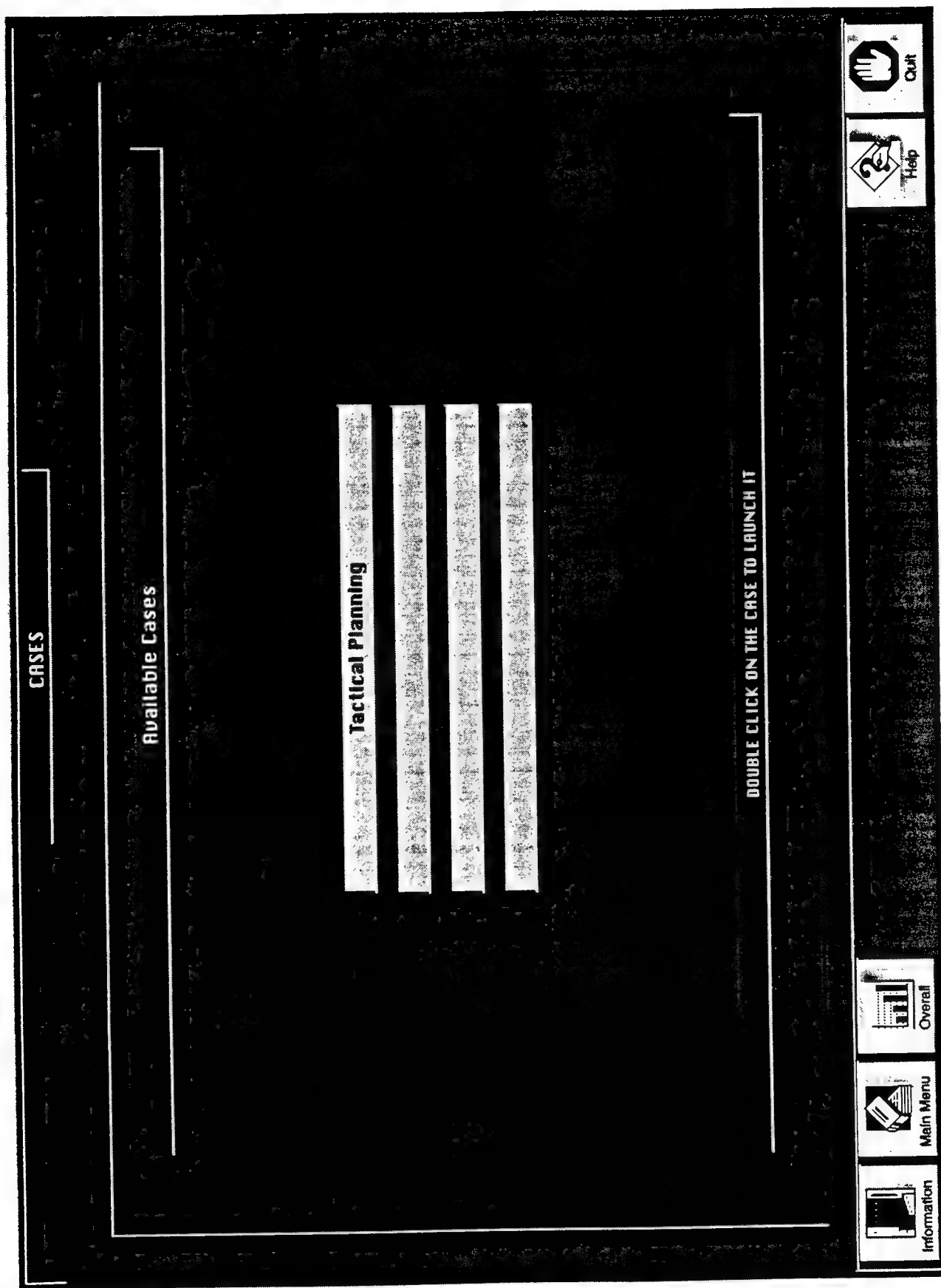


Figure 25: Cases

Figure 26 provides the designer with a status report on how much of the design process has been completed and what remains to be done. If key steps have not been taken then the system will not permit the design process to move to prototyping.

Figure 27 shifts to the display design recommendations that the system makes after requirements, constraints and user profile information has been entered. These recommendations are based on knowledge of the UCI design process. It also indicates how "confident" the system is in the recommendation with a sliding scale that portrays the recommendation as "weak" to "strong."

The system also permits designers to see examples of the recommended UCI features as well as linkages back to the requirements models.

Figure 28 repeats the process for data and information coding recommendations, while Figure 29 presents the dialog and interaction recommendations.

Figure 30 provides prototyping options (by platform). This is the path to the COTS tools that can be used to build interactive prototypes that feature the UCI display and interaction recommendations made by the system.

Figure 31 presents cases to the designer, cases that can help with the development of the prototype.

Figure 32 again checks the status of the design and development process.

Figure 33 permits the designer to recall previous inputs and inspect how they "trace" to features and their representation in the prototype.

Figure 34 walks the designer through the evaluation process.


Figure 35 presents some testing options to the designer.

Figure 36 presents another checklist, while Figure 37 provides another status check.

4.3 The FLEX Tutorial

Embedded in the system is a tutorial based on the FLEX case study. This tutorial is intended to introduce designers to the DesignPro system and to the domain of command and control, the only knowledge domain currently in the system.

OVERALL REQUIREMENTS STATUS


 Task Analysis	 Hierarchy	 Annotations	 Importance	 Not Completed
 Users	 Characteristics	 Annotations		 Not Completed
 Constraints	 Hardware	 Management		 Not Completed


Overall Status:


Proceed/Cannot Proceed to Design & Prototyping


Recommended Design Steps:

>>>Task analysis not yet completed.
Return to this section and attend to the areas indicated.
At least 3 of the first level tasks of the project must be titled and described.
• Level 0 – MISSING


Information


Main Menu


Print Report


Help



Quit

Figure 26: Status Display

DISPLAY DESIGN

Example

Confidence Level

Inclusion

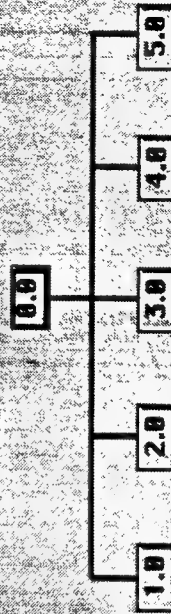
Display Features

FEATURES	Yes	No	Weak	Strong
Analog Displays	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Bar Charts	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Column Charts	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Band Charts	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Status Indications	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Pattern Displays	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Trend Displays	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Form Fill-In	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Windows	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Palettes	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Embedded Menus	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Multiple Windows	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Pop-Down Menus	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Pop-Up Menus	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

DEFINITION FOR:

Analog displays definition.

RATIONALE FOR:



Information



Main Menu



Overall



Data /
Information
Coding



Dialog &
Interaction



Help



Quit

Figure 27: Display Recommendations

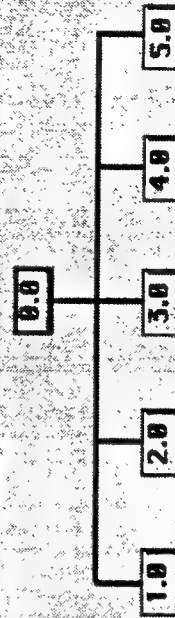
Example

Dialog & Interaction Features Inclusion Confidence Level

FEATURES:	Yes	No	Weak	Strong
Task Automation	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Natural Language	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Hierarchical Menu	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Single Menu	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Command Language	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Hypertext	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
GUI / Iconic	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Command Menu	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Virtual Reality	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Speech	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

DEFINITION FOR:

RATIONALE FOR:



Information



Main Menu



Overall



Data /
Information
Coding



Display Design



Help



Quit

Figure 29: Dialog & Interaction Recommendations

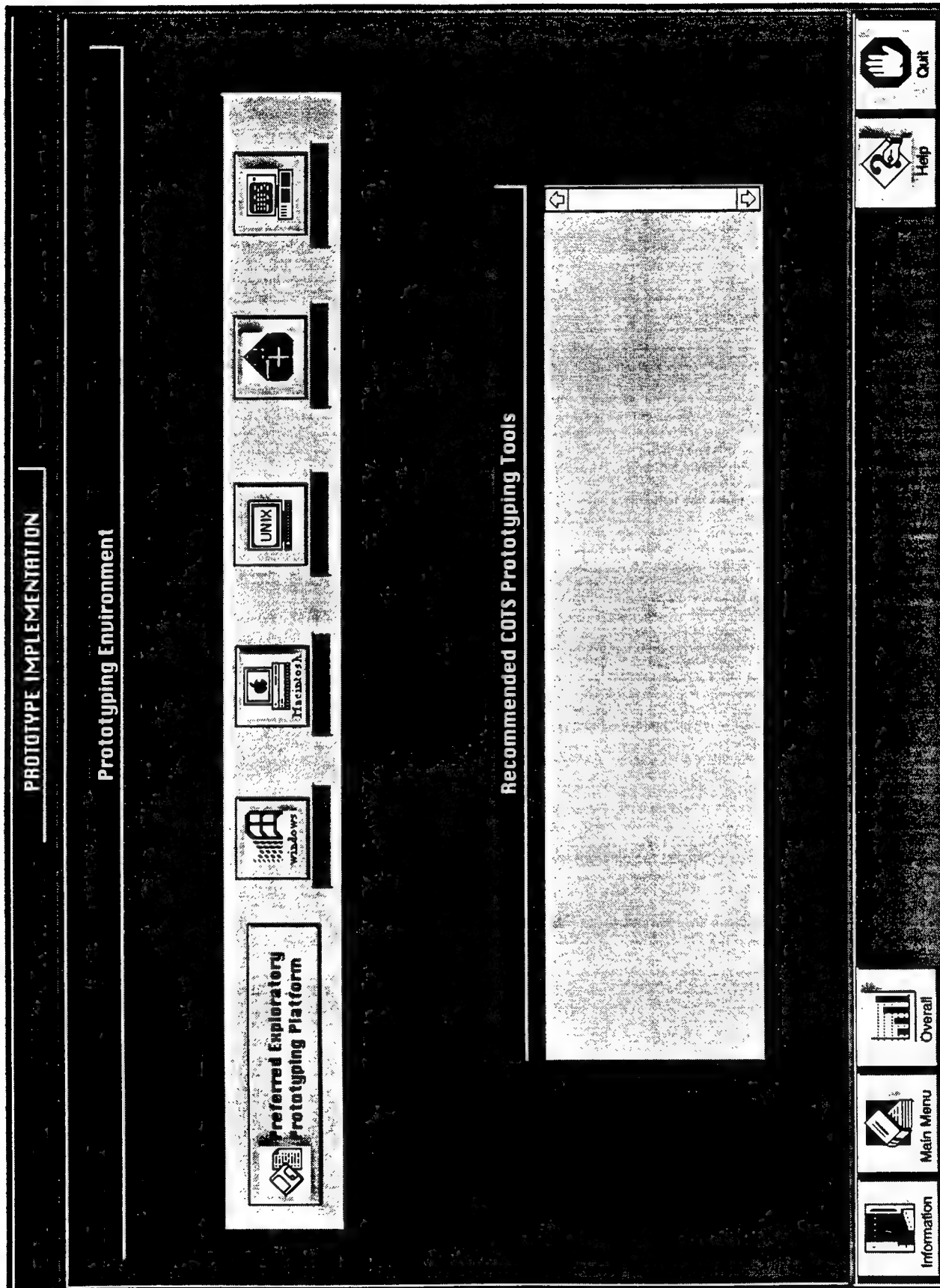


Figure 30: Prototyping Options

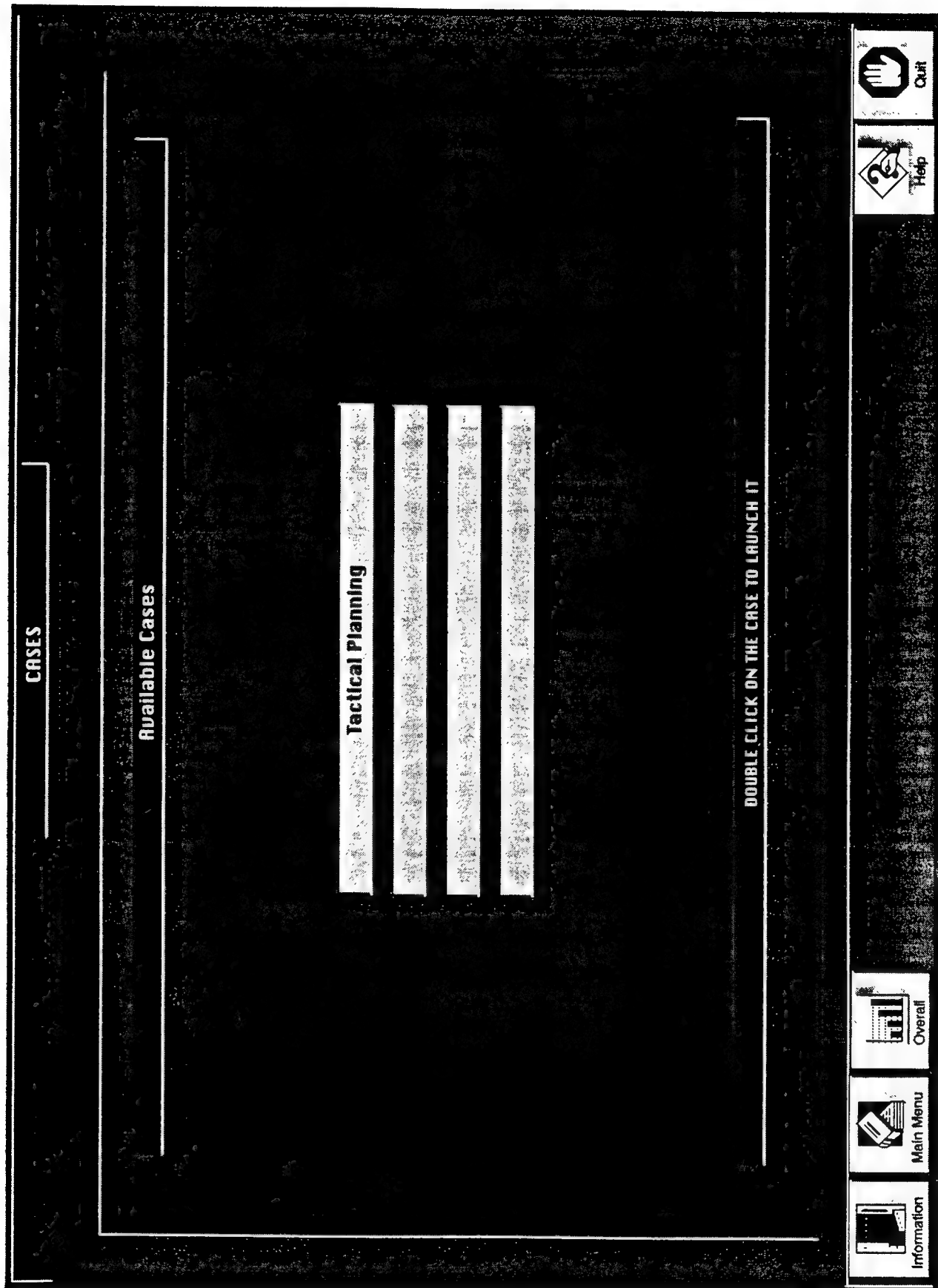


Figure 31: Cases

Display Design

Data /
Information
Coding

Dialog &
Interaction

Implementation



Information



Main Menu



Print



Print All



Quit



Help

Figure 32: Status Check

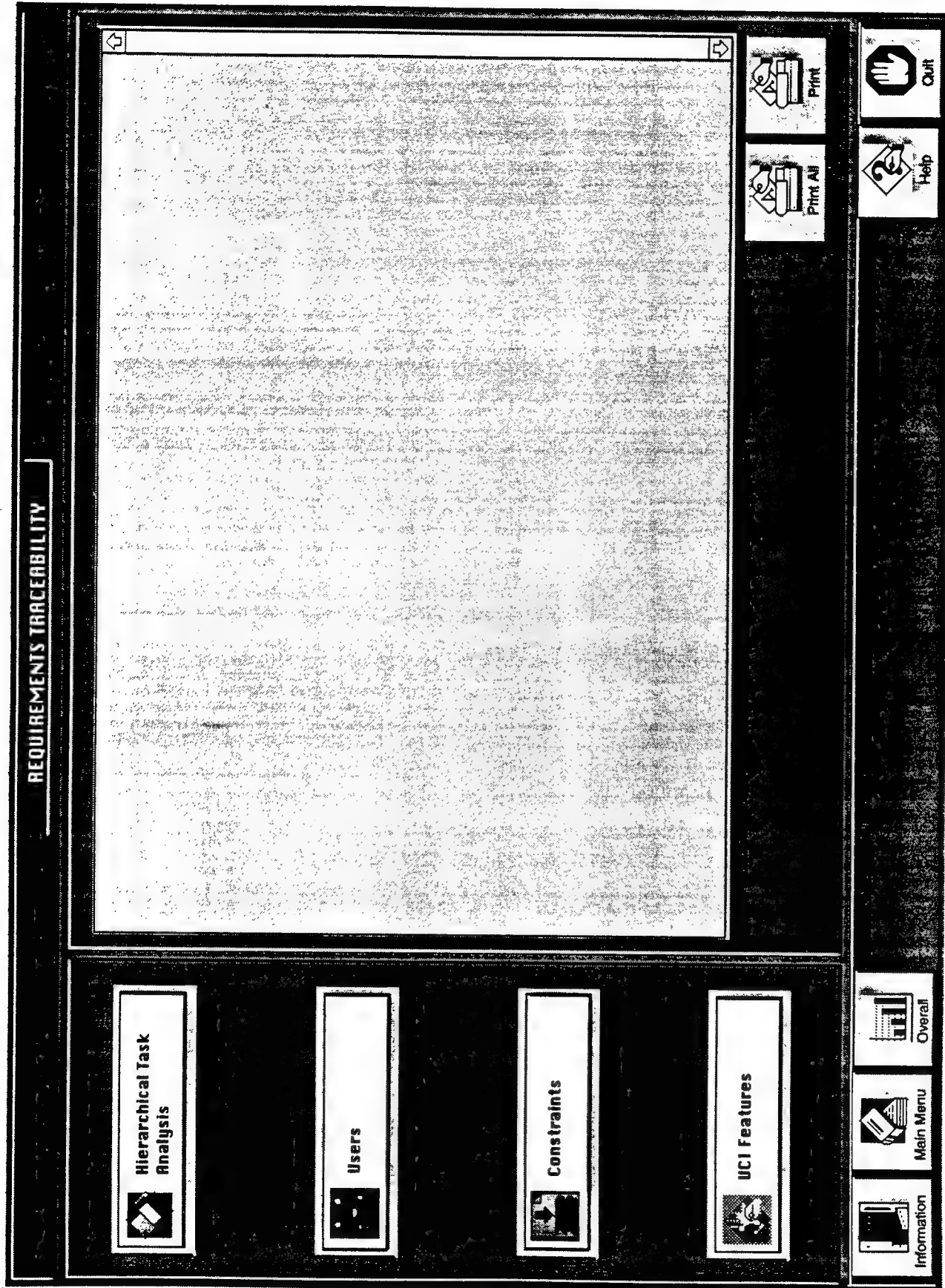


Figure 33: Traceability

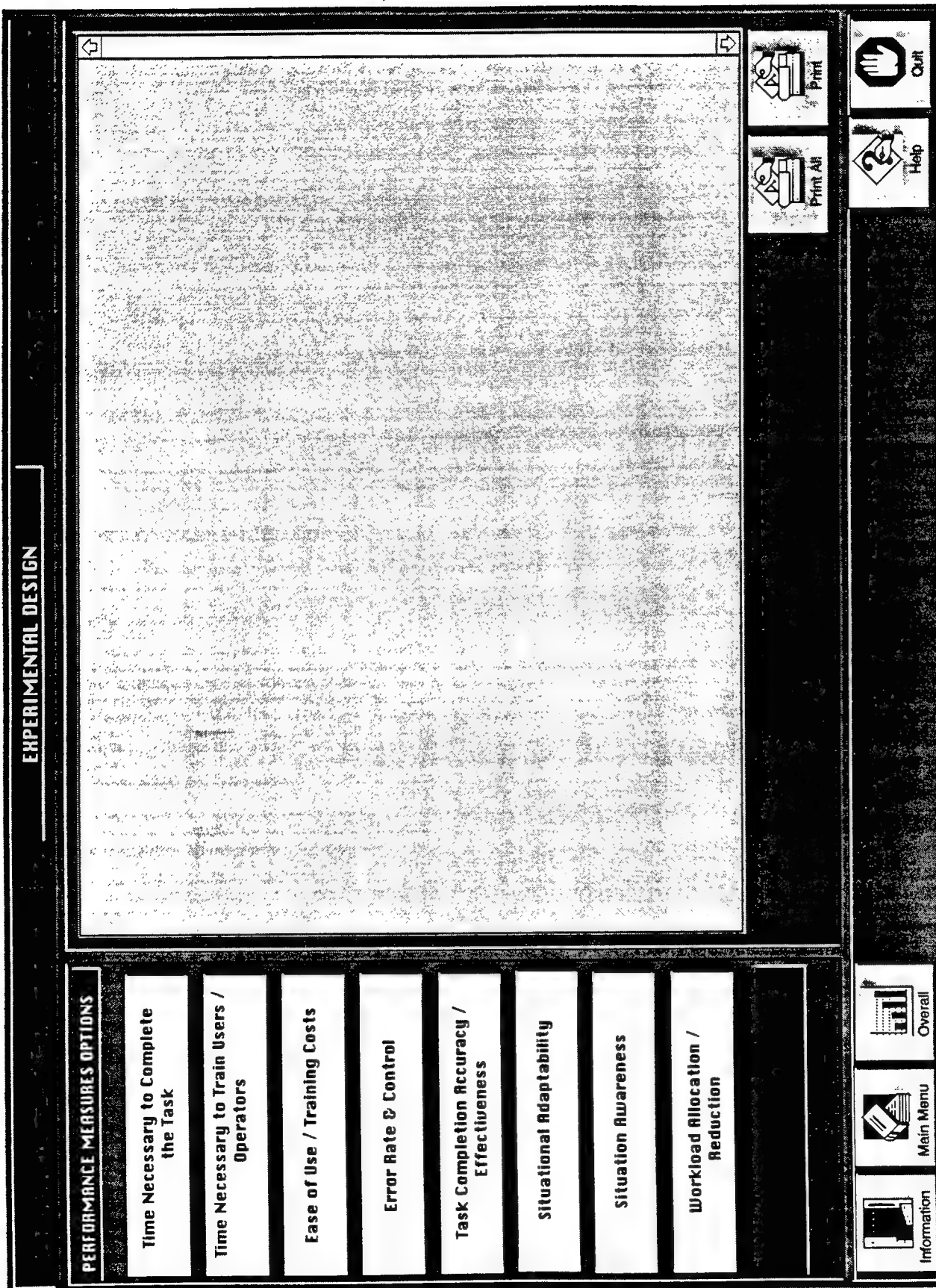


Figure 34: Evaluation Process

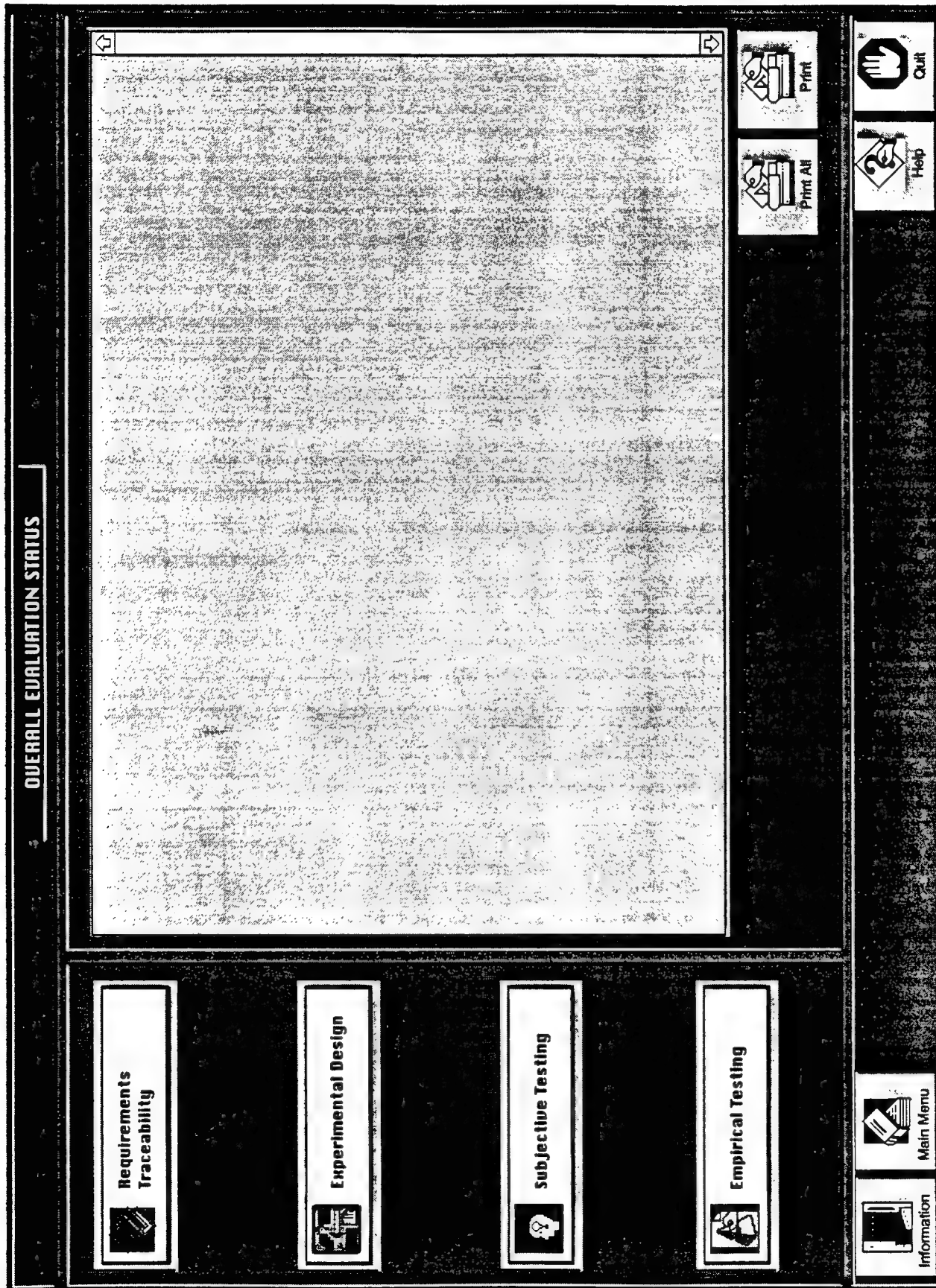


Figure 35: Testing Options

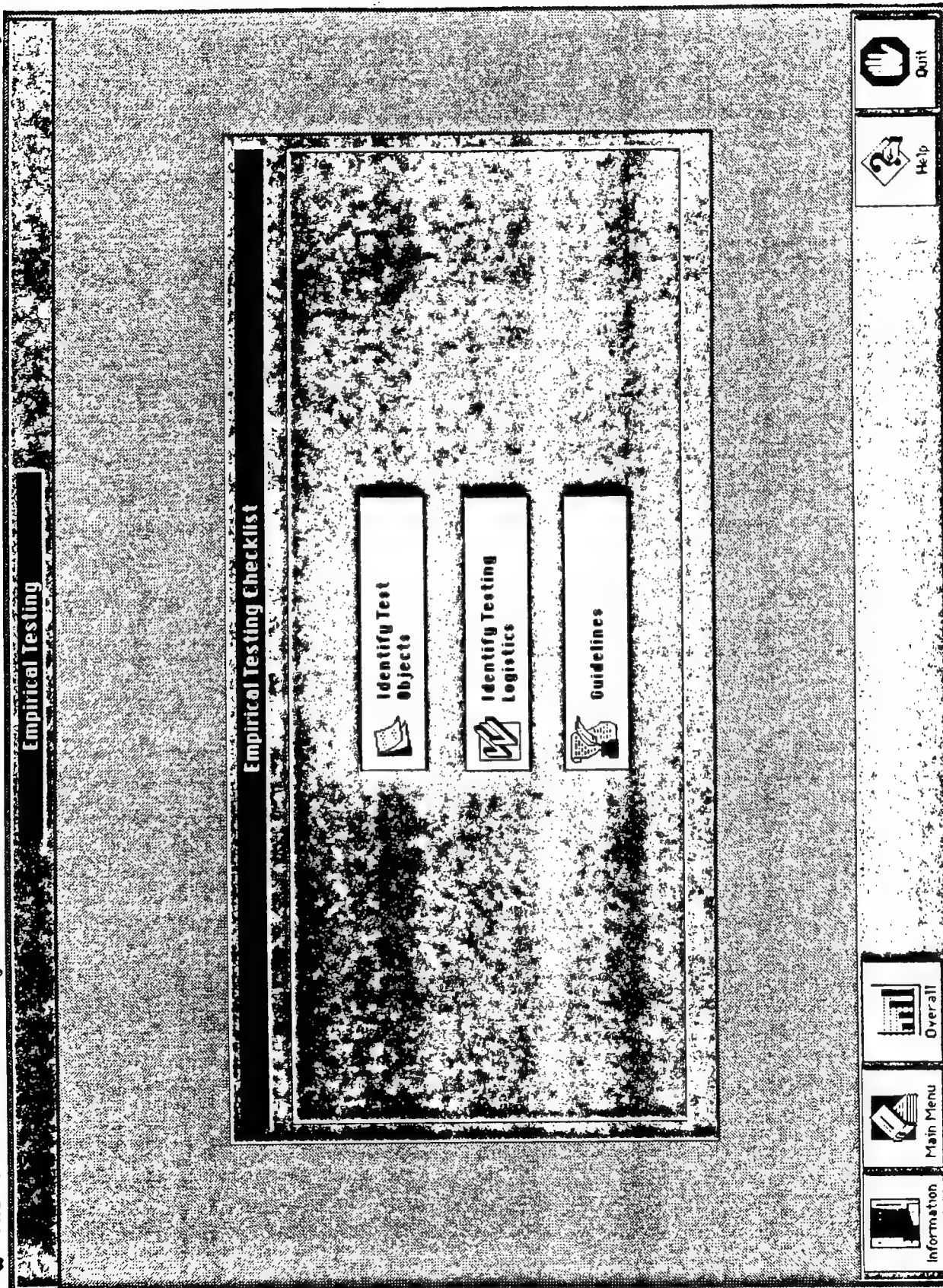


Figure 36: Checklist

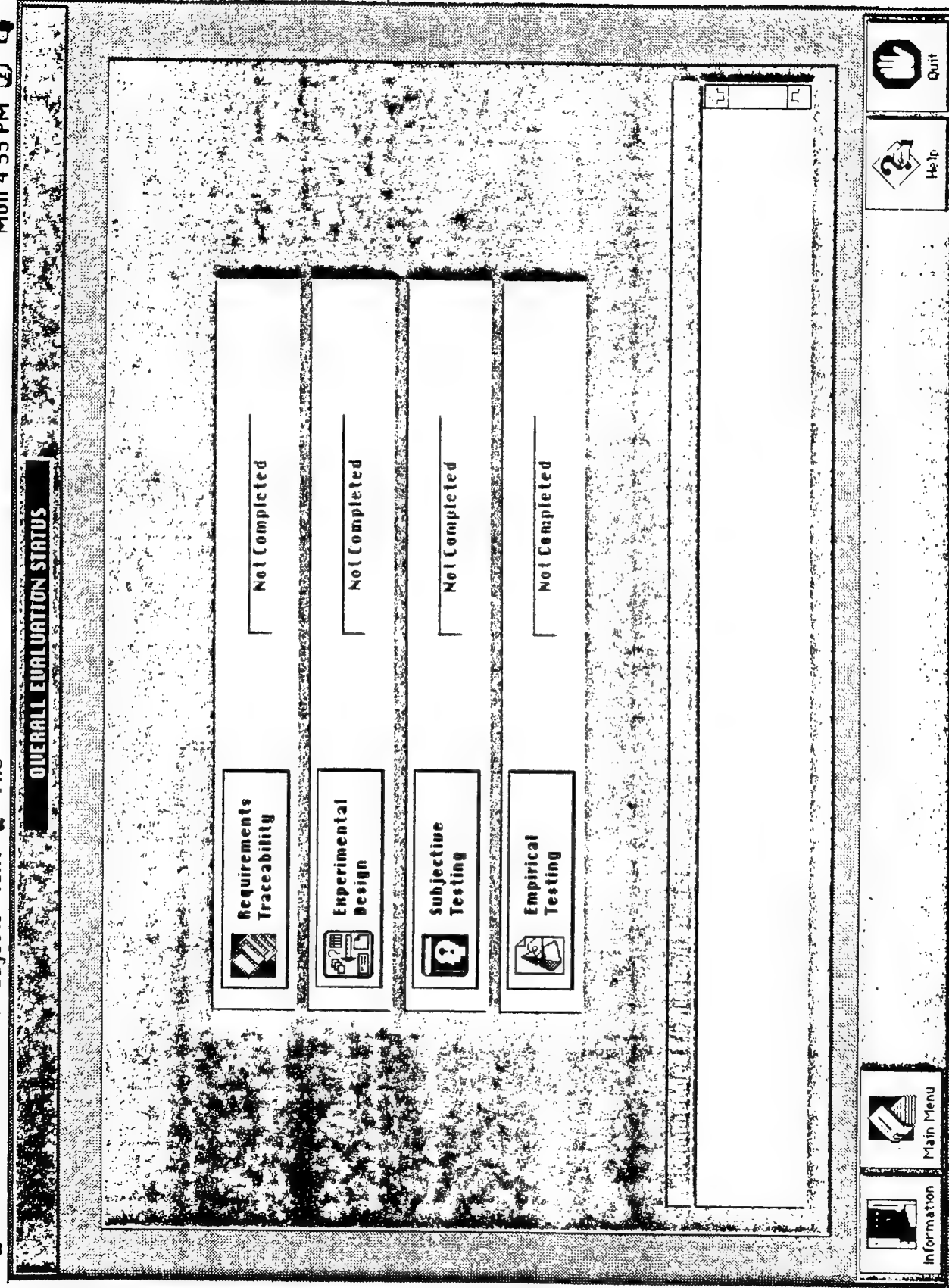


Figure 37: Status Check

5.0 Summary & Conclusions

This final technical report covers the period from August 1, 1992 through August 31, 1995. The report describes progress made in the development of the DesignPro interactive computer-based advisory system for user-computer interface (UCI) design, prototyping and evaluation.

The overall process includes interaction among knowledge templates to develop a requirements model that, in turn, helps yield displays and UCI routines which, in turn, suggest a prototyping strategy which, in turn, identifies evaluation tactics.

The DesignPro system supports to the UCI designer; it does not call for the replacement of human UCI expertise in the design process. The methodology assumes that commercial-off-the-shelf (COTS) software can be used to create (simulate) an integrated environment for designing, prototyping and evaluating interactive user computer interaction routines.

The project was anchored in the systems engineering approach to interactive systems design and development; the throwaway ---> evolutionary prototyping approach to validate requirements was implemented. An initial prototype was released in January of 1993; another in April of 1993 and another in October 1993; refinements were made to the prototype in January and March of 1994 and then again in April 1995, with a final prototype release in August 1995. The prototypes were used to validate workstation design requirements and to communicate what the system does. They also permitted us to integrate many concepts, tools, and COTS software programs into the design.

The project has also pursued a case study within its scope. The FLEX case study was completed during this reporting period and presented to Rome Laboratory personnel in July 1994. FLEX illustrated how the UCI design, prototyping, and evaluation methodology embedded in DesignPro can be used to design, prototype and evaluate varieties of command and control interfaces.

The DesignPro Advisory System permits designers of user computer interfaces to represent requirements, to build prototypes, and to evaluate their impact -- all via a "workbench" of user accessible functions.

The following figure presents the DesignPro workbench. Note the major functional areas and the system's ability to show examples of the features that comprise user computer interfaces as well as examples of off-the-shelf prototyping tools via a "browser" capability.

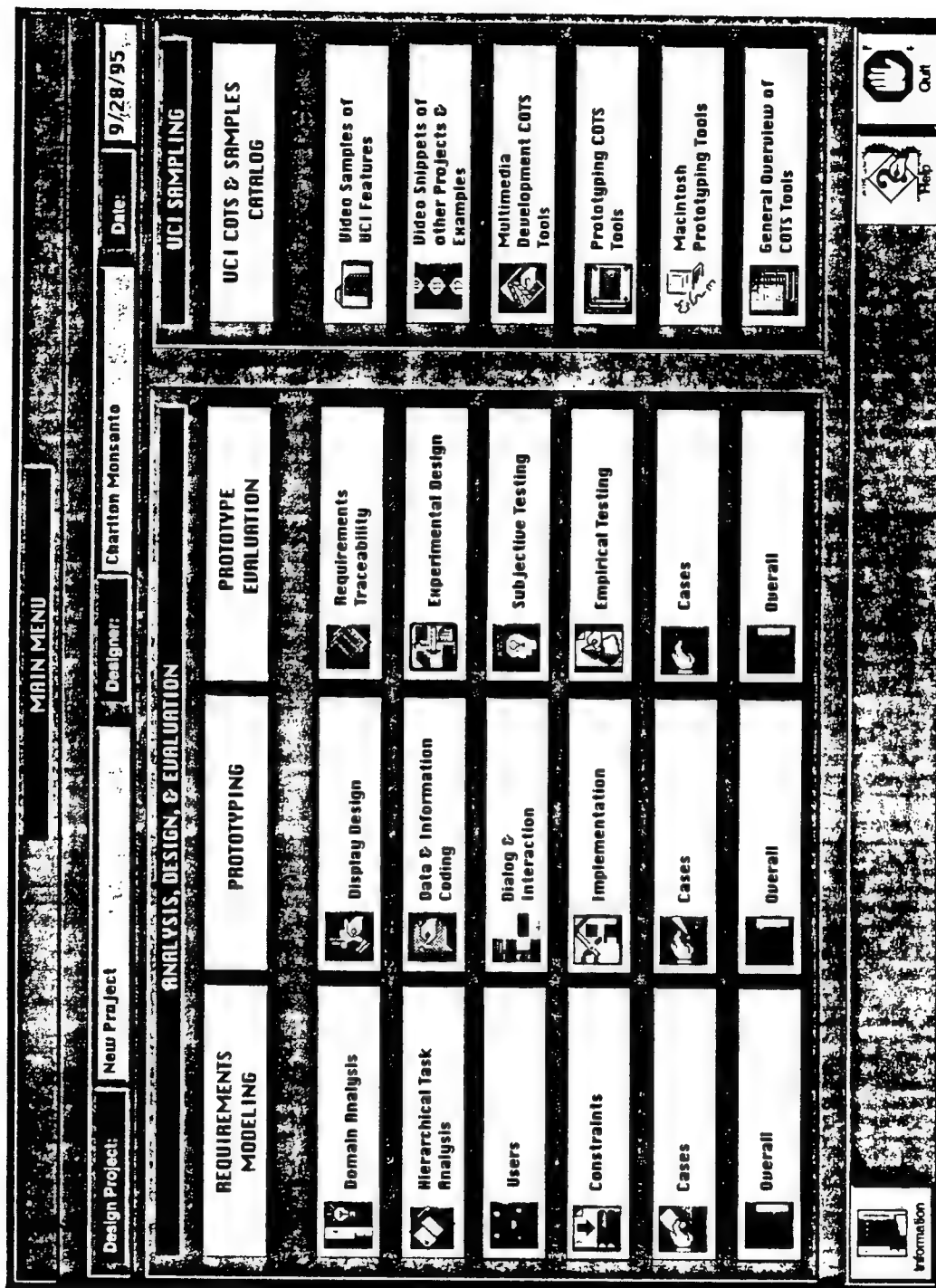


Figure 38: The DesignPro Workbench

The project synthesized findings from a variety of sources and disciplines -- as suggested by the following figure:

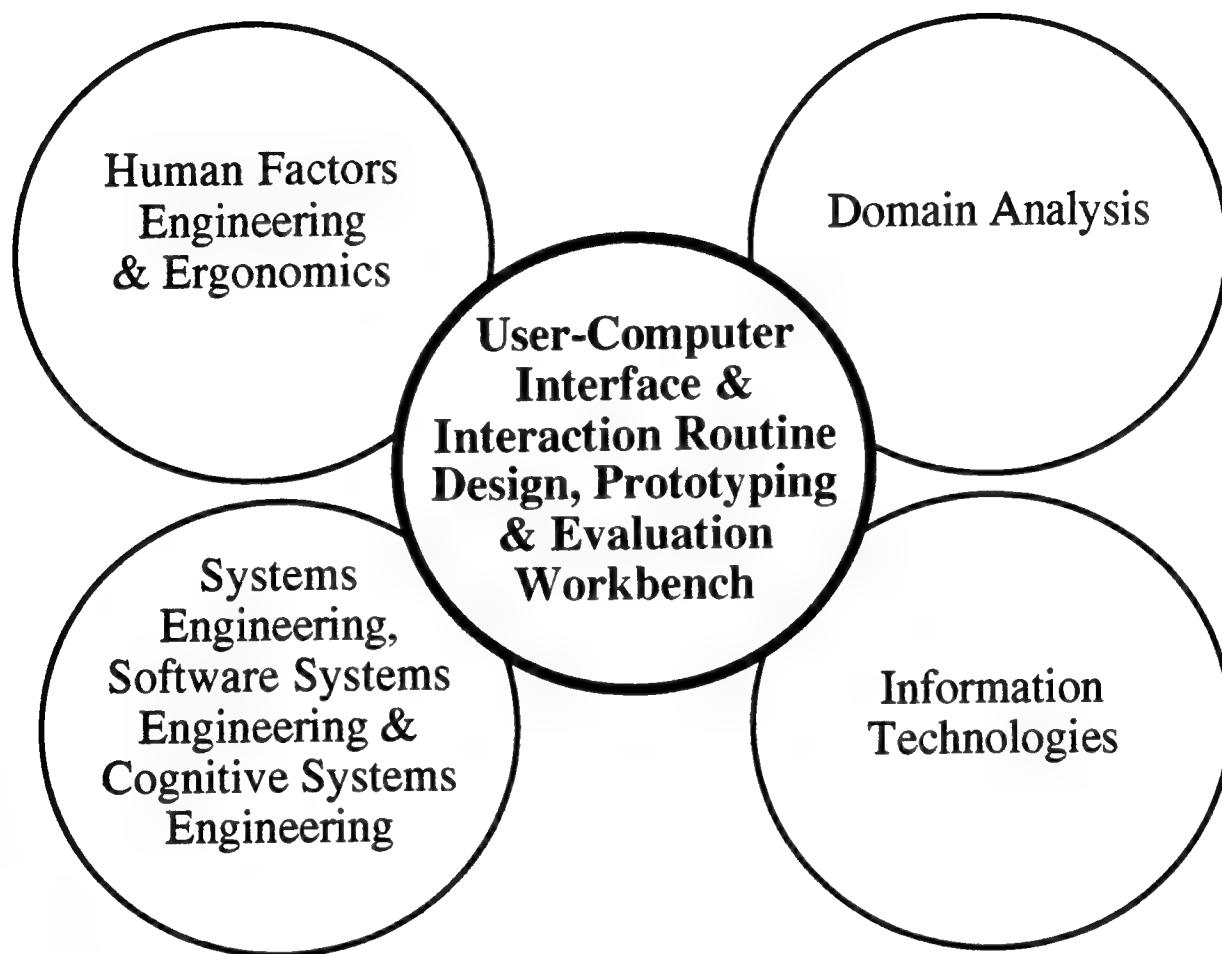


Figure 39: The Project's Analytical Backdrop

The project's ultimate payoff will depend upon the nature of the user-computer interface design applications to which the workbench is applied; a large number of applications will provide insight into the operational capabilities of the interface and the analytical and design assumptions upon which it is based.

Ultimately the workbench demonstrates a growing trend in the design arena: the embedding of more and more design expertise in the knowledge-based software systems capable of -- in most cases -- advising designers and -- in a few cases -- automating design processes.

6.0 References & Bibliography

- ACM. *Proceedings of the 3rd Testing, Analysis and Verification Workshop*, December, 1989.
- Adelman, L., M. Tolcott, and T. Bresnick. (1993). "Examining the Effect of Information Order on Expert Judgment." *Organizational Behavior and Human Processes*, 56, 348-369.
- Agresti, W. *The New Paradigms of Software Development*, IEEE Computer Society, Los Alamitos, CA., 1986.
- Aiken, P.H. and L. S. Ehrhart. (1991). "Making Information in Computerized Databases More Accessible: A Cognitive Engineering Case Study." In *Proc. 29th Annual Technical Symposium. Advanced Information Interfaces: Making Data Accessible, June 20, 1991*. Washington, DC: ACM, 1991, 101-110.
- Allen, R. B. (1982). "Cognitive Factors in Human Interaction with Computers." In A. Badre and B. Shneiderman (Eds.), *Directions in Human/Computer Interaction*. Norwood, NJ: Ablex.
- American Heritage Dictionary*, Second College Edition. (1985). Boston: Houghton Mifflin, 1985.
- Anderson, J. R. (1983). *The Architecture of Cognition*. Cambridge, MA: Harvard University Press.
- Andriole, S. J. (1987). "Storyboard Prototyping for Requirements Verification," *Large Scale Systems*, 12, 1987, 231-247.
- _____. (1989). *Handbook of Decision Support Systems*. Blue Ridge Summit, PA: Tab Books, Inc.
- _____. (1990). *Information System Design Principles for the 90s: Getting it Right!* Fairfax, VA: AFCEA International Press.

- Andriole, S. J. & L. Adelman. (1989). *Cognitive Engineering of Advanced Information Technology for Air Force Systems Design & Development: Prototypes for Air Defense Intelligence & Operations*. Marshall, VA: International Information Systems, Inc.
- Andriole, S.J. and L.S. Ehrhart. (1990). *Intelligent Decision Support for Electronic Warfare: The IDSSEW Prototype*. Fairfax, VA: Center for Excellence in Command, Control, Communications and Intelligence, George Mason University, September 1990.
- Arthur, L. J. (1992). *Rapid Evolutionary Development: Requirements, Prototyping & Software Creation*. New York: John Wiley, 1992.
- Badre, A. N. (1992). "Selecting and Representing Information Structures for Visual Presentation," *IEEE Transactions on Systems, Man, and Cybernetics*, 12(4), 495-504.
- Bailey, R. W. (1989). *Human Performance Engineering: Using Human Factors / Ergonomics to Achieve Computer System Usability*, 2nd Ed. Englewood Cliffs, NJ: Prentice-Hall.
- Bainbridge, L. (1987). "The Ironies of Automation." In J. Rasmussen, K. Duncan & J. Leplat (Eds.), *New Technology and Human Error*. London: Wiley.
- _____. (1988). "Types of Representation." In L. P. Goodstein, H. B. Andersen, and S. E. Olsen (Eds.), *Tasks, Errors, and Mental Models*. London: Taylor & Francis.
- Ball, R. G. and G. Ord. (1983). "Interactive Conflict Resolution in Air Traffic Control." In M. J. Coombs and J. L. Alty (Eds.), *Designing for Human-Computer Communication*. London: Academic Press.
- Banville, C. and Landry, M. Can the field of MIS be disciplined?. *Communications of the ACM*, Vol. 32, (Jan. 1989), pp. 48-60.
- Barclay, et al 1977, *Handbook for Decision Analysis, Decisions & Designs*, Inc., McLean, VA.
- Basili, V.R. and Rombach, H.D. The TAME project: Towards Improvement-Oriented Software environments. *IEEE Transactions on Software Engineering*, June 1988, pp. 728-738.

- Basili, V. & D. Weiss. (1981). "Evaluation of a Software Requirements Document by Analysis of Change Data." In *Fifth IEEE International Conference on Software Engineering*. Washington, DC: Computer Society Press of the IEEE, pp. 314-323.
- Battilega, J. A. and J. K. Grange, (Eds.). (1980). *The Military Applications of Modeling*. Wright-Patterson AFB, Ohio: Air Force Institute of Technology Press, 1980.
- Beach, L. R. (1992). "Epistemic Strategies: Causal Thinking in Expert and Nonexpert Judgment." In G. Wright and F. Bolger (Eds.), *Expertise and Decision Support*. New York: Plenum Press.
- Bennett, K. B. and J. M. Flach. (1992). "Graphical Displays: Implications for Divided Attention, Focused Attention, and Problem Solving." *Human Factors*, 34(5), 513-533.
- Bennett, K. B., M. L. Toms, and D. D. Woods. (1993). "Emergent Features and Graphical Elements: Designing More Effective Configural Displays." *Human Factors*, 35(1), 71-97.
- Berliner, D. C., D. Angell, and J. Shearer. (1964). "Behaviors, Measures and Instruments for Performance Evaluation in Simulated Environments." In *Proc. of the Symposium and Workshop on the Quantification of Human Performance*. University of New Mexico, Albuquerque, NM.
- Bersoff, E. H. (1984). "Elements of Software Configuration Management," *IEEE Transactions on Software Engineering*, Vol. SE-10, No. 1, January 1984, 79-87.
- Blanchard and Fabrycky, 1981, *System Engineering*, Prentice-Hall, Englewood Cliffs, NJ.
- Boar, B. H. (1984). *Application Prototyping: A Requirements Strategy for the 80's*. New York: Wiley-Interscience.
- Boehm, B. W. (1975). "Some Experience with Automated Aids to the Design of Large-Scale Reliable Software." *IEEE Transactions on Software Engineering*, 11(8), March 1975, 125-133.

- Boehm, B. W. Software Engineering. *IEEE Transactions on Computers*. Vol. C-25, No. 12, December, 1976, pp. 1226-1241.
- Boehm, B. W. A Spiral Model of Software Development. *Computer*, May, 1988, pp. 61-72.
- _____. (1976). "Software Engineering," *IEEE Transactions on Computers*, 25(12), December 1976, 1226-1241.
- _____. (1981). *Software Engineering Economics*. Englewood Cliffs, NJ: Prentice-Hall, 1981.
- _____. (1987). "Improving Software Productivity," *IEEE Computer*, September 1987, 43-57.
- Bott, R. (1979) "A Study of Complex Learning: Theory and Methodology," *CHIP Report 82*, La Jolla, CA: Center for Human Information Processing, University of California at San Diego.
- Brehmer, B. (1987). "Models of Diagnostic Judgment." In J. Rasmussen, K. Duncan, and J. Leplat (Eds.), *New Technology and Human Error*. New York: Wiley, 1987.
- Brooke, J.B. and K.D. Duncan. (1981). "Effects of System Display Format on Performance in a Fault Location Task," *Ergonomics*, 1981, 24, 175-189.
- Brooks, F. P. (1988). "Grasping Reality Through Illusion — Interactive Graphics Serving Science," *Proceedings: ACM/SIGCHI Conference on Human Factors in Computing Systems*, May 1988, pp. 1-10.
- Bruggere, T. (1979). "Software Engineering Management, Personnel, and Methodology. In *Fourth IEEE International Conference on Software Engineering*, Washington, DC.: IEEE Computer Society Press.
- Byrd, T. A., K. L. Cossick, and R. W. Zmud. (1992). "A Synthesis of Research on Requirements Analysis and Knowledge Acquisition Techniques." *MIS Quarterly*, March 1992, 117-138.
- Card, S. K., T. P. Moran, and A. Newell. (1983). *The Psychology of Human-Computer Interaction*. Hillsdale, NJ: Lawrence Erlbaum.

- Carroll, J. M. and R. L. Campbell. (1988). *Artifacts as Psychological Theories: The Case of Human-Computer Interaction*. [Technical Report RC 13454 (#60225)]. Yorktown Heights, NY: IBM Research Division, T. J. Watson Research Center.
- Carroll, J. M., R. L. Mack and W. A. Kellogg. (1988). "Interface Metaphors and User Interface Design." In M. Helander (Ed.) *Handbook of Human-Computer Interaction*. New York: Elsevier.
- Carroll, J. M. and J. R. Olson. (1988). "Mental Models in Human-Computer Interaction." In M. Helander (Ed.), *Handbook of Human-Computer Interaction*. Amsterdam: Elsevier.
- Carswell, C. M. and C. D. Wickens. (1987). "Information Integration and the Object Display." *Ergonomics*, 30, 511-527.
- Castellan, N. J., Jr. (Ed.) (1993) *Individual and Group Decision Making*. Hillsdale, NJ: Lawrence Erlbaum.
- Chambers, N. and H. Nagel. (1985). "Computer-Aided Pilots." *Communications of the ACM*, November.
- Chambers, G.J., The System Engineering Process: a Technical Bibliography. *IEEE Transactions on Systems, Man & Cybernetics*, September/ October 1986, Vol. SMC-16, No. 5, pp. 712-721.
- Chapanis, A. (1959). *Research Techniques in Human Engineering*. Baltimore, MD: Johns Hopkins Press.
- Chestnut, H. *Systems Engineering Methods*. Wiley, New York, NY, 1967.
- Cohen, M. S., B. B. Thompson, and J. O. Chinnis, Jr. (1985). *Design Principles for Personalized Decision Aiding: an Application to Tactical Air Force Route Planning*. RADC-TR-85-140. Final Technical Report. McLean, VA: Decision Science Consortium, Inc., July.
- Connell, J. L. and L. Shafter. (1989). *Structured Rapid Prototyping: An Evolutionary Approach to Software Development*. Englewood Cliffs, NJ: Prentice Hall.

- Coury, B. G. and M. D. Boulette. (1992). "Time Stress and the Processing of Visual Displays." *Human Factors*, 34(6), 707-725.
- Daly, E. (1977). "Management of Software Development." *IEEE Transactions on Software Engineering*, 3(3), May, 229-242.
- Davis, A. M. (1993). *Software Requirements: Objects, Functions, and States*. Revised Ed. Englewood Cliffs, NJ: Prentice-Hall.
- Davis, Alan M., *et al.* A Strategy for Comparing Alternative Software Development Life Cycle Models. *IEEE Transactions on Software Engineering*, 14,10 (October, 1988), pp. 1453-61.
- Department of Defense (DOD). *Engineering Management, MIL-STD-499A (USAF)*, May 1, 1974, pp. 11-18.
- Defense Systems Management College (DSMC), *Systems Engineering Management Guide*, Washington, DC, Fort Belvoir, 1986.
- Department of Defense (DOD). *Engineering Management, MIL-STD-499A (USAF)*, May 1, 1974, pp. 11-18.
- Department of Defense (DOD). (1992). *HCI Style Guide Version 2.0. Final Draft*. Washington, D.C.: Department of Defense, September.
- diSessa, A. A. (1988). "Knowledge in Pieces." In G. Forman and P. B. Pufall (Eds.), *Constructivism in the Computer Age*. Hillsdale, NJ: Erlbaum.
- Dörner, D. (1987). "On the Difficulties People Have in Dealing with Complexity." In J. Rasmussen, K. Duncan and J. Leplat (Eds.), *New Technology and Human Error*. New York: John Wiley, 1987.
- Dreyfus, J. L. and S. E. Dreyfus. (1986). *Mind Over Machine: The Power of Human Intuition and Expertise in the Era of the Computer*. New York: Free Press.
- Duffy, L. (1993). "Team Decision-Making and Technology." In N. J. Castellan, Jr. (Ed.), *Individual and Group Decision Making*. Hillsdale, NJ: Erlbaum.
- Edgell, S. E., R. D. Bright, P. C. Ng, T. K. Noonan, and L. A. Ford (1992). "The Effects of Representation on the Processing of Probabilistic Information." In B. Burns (Ed.), *Percepts, Concepts and Categories: The*

Representation and Processing of Information. Amsterdam: Elsevier Science Publishers.

Edland, A & O. Svenson. (1993). "Judgment and Decision Making under Time Pressure: Studies and Findings." In O. Svenson & A. J. Maule (Eds.), *Time Pressure and Stress in Human Judgment and Decision Making*. New York: Plenum Press, 27-40.

Ehrhart, L.S. (1990). "New Approaches to Human-Computer Interaction Research and Design for Decision Aiding Systems." In *Proc. IEEE International Intelligent Control Symposium*, Philadelphia, PA, 5-7 September.

_____. (1993a). "Integrating Multimedia Information for Theater-Level Command and Control." Final Report. Air Force Office of Scientific Research, Research Development Laboratories, Summer Research Program. Washington, DC: AFOSR. September 1993.

_____. (1993b). *Tasks, Models & Methods: System-Level Evaluation of Human-Computer Interaction (HCI) for Command & Control Decision Support*. (R&D Interim Technical Report #MISE-3-1-3. Submitted to Rome Laboratory/C³AA under Contract #F30602-92-C-0119). Fairfax, VA: C³I Center, George Mason University, 1 March 1993.

Ehrhart, L.S. and P.H. Aiken. (1991). "Cognitive Engineering for Intelligent Control System Design: Preserving User Models in Requirements Analysis." In *Proc. of the 1991 American Control Conference*. Boston, MA, 26-28 June, 1991. Evanston, IL: American Automatic Control Council.

Eisner, H., *Computer-Aided Systems Engineering*, Prentice-Hall, Englewood Cliffs, NJ, 1988.

Elm, W. C. & D. D. Woods. (1985). "Getting Lost: A Case Study in Interface Design." In *Proceedings of the Human Factors Society*, 29th Annual Meeting.

Fagan, M. (1974). *Design and Code Inspections and Process Control in the Development of Programs*. IBM Report IBM-SDD-TR-21-572.

December.

- Fleishman, E. A. & M. K. Quaintance. (1984). *Taxonomies of Human Performance: The Description of Human Tasks*. New York: Academic Press.
- Freeman, P. *Software Perspectives*, Addison-Wesley, Reading, MA, 1987.
- Freeman, P. and von Staa, A. *Towards a Theory of Software Engineering*. Working paper, November, 1984, available from PAF.
- Freeman, P., Wasserman, A. I., and Fairley, R. E. Essential Elements of Software Engineering Education. *Proc. 2nd International Conference on Software Engineering*, IEEE Computer Society, Los Alamitos, CA., 1976, pp. 116-122.
- French, W. L. and C. H. Bell, Jr. (1973). *Organization Development: Behavioral Science Interventions for Organization Improvement*. Englewood Cliffs, NJ: Prentice-Hall.
- Gardiner, M. M. & B. Christie. (1987). *Applying Cognitive Psychology to User-Interface Design*. New York: John Wiley & Sons.
- Gentner, D. and D. R. Gentner. (1983). "Flowing Waters or Teeming Crowds: Mental Models of Electricity." In D. Gentner and A. L. Stevens (Eds.), *Mental Models*. Hillsdale, NJ: Lawrence Erlbaum.
- Gilchrist, A. (1990). "Perception of Surface Blacks and Whites." In I. Rock (Ed.), *The Perceptual World*. New York: W. H. Freeman.
- Goldsmith, T.E. and R. W. Schvaneveldt. (1984). "Facilitating Multiple-Cue Judgements with Integral Displays." In J. C. Thomas & M. L. Schneider (Eds.), *Human Factors in Computing Systems*. Norwood, NJ: Ablex.
- Goode, H. and Machol, R.E. *Systems Engineering: An Introduction to the Design of Large Scale Systems*, McGraw-Hill, New York, NY, 1957.
- Gowan, D. B. & J. D. Novak. (1984). *Learning to Learn*. New York: Cambridge University Press.

- Green, T. R. G. (1990). "Limited Theories as a Framework for Human-Computer Interaction." In D. Ackerman and M. J. Tauber (Eds.), *Mental Models and Human-Computer Interaction I*. Amsterdam: Elsevier.
- Hall, A.D. *Methodology for Systems Engineering*, Von Nostrand, Princeton, NJ, 1962.
- Hammond, K. R. (1993). "Naturalistic Decision Making From a Brunswikian Viewpoint: Its Past, Present, Future." In G. A. Klein, J. Orasanu, R. Calderwood, and C. E. Zsombok (Eds.), *Decision Making in Action: Models and Methods*. Norwood, NJ: Ablex.
- Harris, J. F. (1994). "Downing of Copters Blamed on Blunders." *Washington Post*, 117(221), July 14: A1, A10.
- Harrison, T. M. (1985). "Communication and Participative Decision Making : An Exploratory Study." *Personnel Psychology*, Spring, 38(1), p. 93-116.
- Hart, S. G. & L. Staveland. (1988). "Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In P. A. Hancock and N. Meshkati, *Human Mental Workload*. New York: North-Holland, 139-183.
- Helmreich, R. L. (1988). *Testimony before the U.S. House of Representatives, Committee on Armed Services on the Subject of the USS VINCENNES Downing of Iranian Air Flight 655*. October 6, 1988.
- Hermann, C. F. (1972). "Threat, Time, and Surprise: A Simulation of International Crisis." In Hermann, Charles F., (Ed.), *International Crises: Insights from Behavioral Research*. New York: Free Press.
- Hill, S., H. P. Iavecchia, J. C. Byers, A. C. Bittner, A. L. Zaklad, R. E. Christ. (1992). "Comparison for Four Subjective Workload Rating Scales." *Human Factors*, 1992, 34(4), 429-439.
- Hoc, J.-M. (1989). "Strategies in Controlling a Continuous Process with Long Response Latencies: Need for Computer Support to Diagnosis."

- Hoerber, F. P. (1981). *Military Applications of Modeling; Selected Case Studies*. New York: Gordon & Breach Science Publishers.
- Hogarth, R. (1987). *Judgement and Choice: the Psychology of Decision*, 2nd Edition. New York: John Wiley.
- Humphrey, Watts S. *Managing the Software Process*. Addison-Wesley, Reading, MA., 1989.
- Iivari, J. and Koskela, E. The PICO Model for IS design. *MIS Quarterly*, 11,3, 1987.
- Irving, S., P. Polson, and J. E. Irving. (1994). "A GOMS Analysis of the Advanced Automated Cockpit." In *Proceedings of CHI, 1994* (Boston, MA, April 24-28, 1994). New York: Association of Computing Machinery.
- Jensen, R. and Tonies, C.C. *Software Engineering*, Prentice-Hall, Englewood Cliffs, N.J. 1979.
- Janis, I. L. (1972). *Victims of Groupthink*. Boston: Houghton Mifflin.
- _____. (1989). *Crucial Decisions: Leadership in Policy Making and Crisis Management*. New York: Free Press.
- Johnson, P., D. Diaper, and J. Long. (1984). "Tasks, Skills and Knowledge: Task Analysis for Knowledge Based Descriptions." In *Interact '84*, Vol. 1. London: IFIP.
- Johnson-Laird, P. N. (1983). *Mental Models*. Cambridge, UK: Cambridge University Press.
- Kaempf, G. L., S. P. Wolf, M. L. Thordsen, & G. A. Klein. (1992). *Decision Making in the AEGIS Combat Information Center*. [Technical Report prepared for Naval Command, Control and Ocean Surveillance Center, San

Diego, CA (Contract N66001-90-C-6023). Fairborn, OH: Klein Associates, Inc.

Kahan, J. P., D. R. Worley, and C. Stasz. (1989). *Understanding Commander's Information Needs*. [Technical Report R-3761-A]. Santa Monica, CA: The RAND Corporation.

Kantowitz, B. H. (1987). "Mental Workload." In P. A. Hancock (Ed.), *Human Factors Psychology*. Amsterdam: Elsevier Science Publishers, 81-121.

Kerola, P. and Freeman, P. A Comparison of Lifecycle Models. *Proc. 5th International Conference on Software Engineering*, IEEE Computer Society, Los Alamitos, CA, 1981, pp.90-99.

T. S. Kuhn, *The Structure of Scientific Revolutions*, 2nd edition, University of Chicago Press, Chicago, 1970.

Kieras, D. E. (1988). "Towards a Practical GOMS Model Methodology for User Interface Design." In M. Helander (Ed.), *Handbook of Human-Computer Interaction*. Amsterdam: Elsevier.

_____. (1992). "Diagrammatic Displays for Engineered Systems: Effects on Human Performance in Interacting with Malfunctioning Systems." *International Journal of Man-Machine Studies*, 36, 861-895.

Kieras, D. E and S. Bovair. (1986). "The Acquisition of Procedures from Text: A Production-System Analysis of Transfer of Training." *Journal of Memory and Language*, 25, 507-524.

Kieras, D. E. and P. G. Polson. (1985). "An Approach to the Formal Analysis of User Complexity," *International Journal of Man-Machine Studies*, 1985, 22, 365-394.

Klapp, S. T. (1987). "Short-Term Memory Limits in Human Performance." In P. A. Hancock (Ed.), *Human Factors Psychology*. Amsterdam: North-Holland.

Klein, G. A. (1989a). "Do Decision Biases Explain Too Much?" *Human Factors Society Bulletin*, 35(5), 1-3.

_____. (1989b). "Recognition-Primed Decisions." In W. B. Rouse (Ed.), *Advances in Man-Machine Systems Research*, Vol. 5. Greenwich, CT: JAI.

_____. (1993a). "A Recognition-Primed Decision (RPD) Model of Rapid Decision Making." In G. A. Klein, J. Orasanu, R. Calderwood, and C. E. Zsombok (Eds.), *Decision Making in Action: Models and Methods*. Norwood, NJ: Ablex.

_____. (1993b). *Naturalistic Decision Making: Implications for Design*. Technical Report: CSERIAC SOAR 93-1. Wright-Patterson AFB, OH: Crew System Ergonomics, Information Analysis Center, April.

Klein, G. A., R. Calderwood & D. MacGregor. (1989). "Critical Decision Method for Eliciting Knowledge," *IEEE Transactions on Systems, Man and Cybernetics*, 19(3), 462-472.

Klein, G. A., J. Orasanu, R. Calderwood, and C. E. Zsombok, (Eds.). (1993). *Decision Making in Action: Models and Methods*. Norwood, NJ: Ablex.

Kleinmuntz, D. N. (1987). "Human Decision Processes: Heuristics and Task Structure." In P. A. Hancock (Ed.), *Human Factors Psychology*. Amsterdam: North-Holland.

Klinger, D. W., S. J. Andriole, L. G. Militello, L. Adelman, G. Klein. (1993). *Designing for Performance: A Cognitive Systems Engineering Approach to Modifying an AWACS Human-Computer Interface*. Final Technical Report. Contract No. F33615-90-C-0533. Fairborn, OH: Klein Associates, Inc., March.

Leddo, J. M., J. O. Chinnis, M. S. Cohen, and F. M. Martin. (1989). "Tactical Decision Making Under Uncertainty and Time Stress." In *Proceedings of the 1989 Symposium on Command and Control Research*, June 27-29, 1989, National Defense University, Ft. Lesley J. McNair, Washington, DC. McLean, VA: SAIC, December.

- Lehman, M. M. Programs, Life Cycles, and Laws of Software Evolution. *Proceedings of the IEEE*, 68,9, (September, 1980), pp. 1060-1061.
- Lehman, M.M., Stenning, V., and Turski, W.M. Another Look at Software Design Methodology. *ACB SIGSOFT Software Engineering Notes* 9, 2 (April, 1984), pp. 38-53.
- Lehner, P. E. and D. A. Zirk. (1987). "Cognitive Factors in User/Expert-System Interaction." *Human Factors*, 29(1), 97-109.
- Leiser, B. (1992). "The Presence Phenomenon and Other Problems of Applying Mental Models to User Interface Design and Evaluation." In Y. Rogers, A. Rutherford & P. A. Bibby (Eds.), *Models in the Mind: Theory, Perspective and Application*. New York: Academic Press.
- Levis, A. H., N. Moray, and B. Hu. (1994). "Task Decomposition and Allocation Problems and Discrete Event Systems." *Automatica*, 30(2).
- Lockhead, G. R. (1992). "On Identifying Things: A Case for Context." In B. Burns (Ed.), *Percepts, Concepts and Categories: The Representation and Processing of Information*. Amsterdam: Elsevier Science Publishers.
- Loftus, G.R. and Mackworth, N.H. (1978). "Cognitive Determinants of Fixation Location During Picture Viewing." *Journal of Experimental Psychology: Human Perception and Performance*. 4, 565-572.
- MacMillan, J. & E. B. Entin. (1991). "Decision-Oriented Display Design and Evaluation." In *Proceedings of the 1991 Symposium on Command and Control Research*, June 24-25, 1991, National Defense University, Ft. Lesley J. McNair, Washington, DC. McLean, VA: SAIC, December.
- Mantei, M. M. and T. J. Teorey. (1988). "Cost/Benefit Analysis for Incorporating Human Factors in the Software Lifecycle." *Communications of the ACM*, 31(4), April, 428-439.
- Maule, A. J. & O. Svenson. (1993). "Theoretical and Empirical Approaches to Behavioral Decision Making and Their Relation to Time Constraints." In

O. Svenson and A. J. Maule (Eds.), *Time Pressure and Stress in Human Judgment and Decision Making*. New York: Plenum Press, 3-25.

Mayer, R.E. (1979) "Can Advance Organizers Influence Meaningful Learning?" *Review of Educational Research*, 49, 371-383.

Mayer, R.E. and B. K. Bromage. (1980). "Different Recall Protocols for Technical Texts Due to Advance Organizers," *Journal of Educational Psychology*, 72, 209-225.

McCormick, E. J. (1976). "Job and Task Analysis." In M.D. Dunnette, (Ed.), *Handbook of Industrial and Organizational Psychology*. Chicago: Rand McNally.

McFarren, M. R. (1987). *Using Concept Mapping to Define Problems and Identify Key Kernels During the Development of a Decision Support System*. Master Thesis. Wright Patterson AFB, OH: Air Force Institute of Technology, School of Engineering.

Meister, D. (1981). *Behavioral Research and Government Policy: Civilian and Military R & D*. New York: Pergamon Press.

_____. (1985). *Behavioral Analysis and Measurement Methods*. New York: John Wiley.

_____. (1991). *Psychology of System Design*. Amsterdam: Elsevier.

Mills, H. D. Principles of Software Engineering. *IBM Systems Journal*, 19,4 (1980), pp. 414-420.

Metersky, M. L. (1993). "A Decision-Oriented Approach to System Design and Development." *IEEE Transactions on Systems, Man and Cybernetics*, 23(4), July/August, 1024-1037.

Montazemi, A. R. and D. W. Conrath. (1986). "The Use of Cognitive Mapping

for Information Requirements Analysis," *MIS Quarterly*, 10(1), March.

National Academy Press. *Scaling Up: A Research Agenda for Software Engineering*. Report of the Computer Science and Technology Board on a workshop held February 13-15, 1989.

NSF. *Workshop on Software Artifact Research*. Preprints of Meeting held January 23-24, 1990, Atlanta, Georgia.

Naur, P., and Randall, B. (eds). *Software Engineering* (report on a meeting held October, 1968). NATO Science Committee, 1969.

Nielsen, J. (1993). *Usability Engineering*. New York: Academic Press.

Nielsen, J. and R. Molich. (1990). "Heuristic Evaluation of User Interfaces." *Proc. of ACM CHI '90 Conference*. (Seattle, WA, 1-5 April 1990), 249-256.

Nielsen, J. and V. L. Phillips. (1993). "Estimating the Relative Usability of Two Interfaces: Heuristic, Formal, and Empirical Methods Compared." In *Proceedings of*

Norman, D. A. "Some Observations on Mental Models." In D. Gentner and A. L. Stevens (Eds.), *Mental Models*. Hillsdale, NJ: Lawrence Erlbaum, 1983.

_____. (1986). "Cognitive Engineering." In D. A. Norman and S. W. Draper (Eds.), *User-Centered System Design: New Perspectives on Human-Computer Interaction*. Hillsdale, NJ: Lawrence Erlbaum.

Norman, D. A. and S. W. Draper (Eds.). (1986). *User-Centered System Design: New Perspectives on Human-Computer Interaction*. Hillsdale, NJ: Lawrence Erlbaum.

Norman, K. L., L. J. Weldon, and B. Shneiderman. (1986). "Cognitive Layouts of Windows and Multiple Screens for User Interfaces," *International Journal of Man-Machine Studies*, 25, 229-248.

- Nygren, T. E. (1991). "Psychometric Properties of Subjective Workload Measurement Techniques: Implications for Their Use in the Assessment of Perceived Mental Workload." *Human Factors*, 1991, 33(1), 17-33.
- O'Malley, C. & S. Draper. "Representation and Interaction: Are Mental Models All in the Mind?" In Y. Rogers, A. Rutherford & P. A. Bibby (Eds.), *Models in the Mind: Theory, Perspective and Application*. New York: Academic Press, 1992.
- Orasanu, J. and E. Salas. (1993). "Team Decisionmaking in Complex Environments." In G. A. Klein, J. Orasanu, R. Calderwood, and C. E. Zsombok (Eds.), *Decision Making in Action: Models and Methods*. Norwood, NJ: Ablex.
- Patrick, J. (1987). "Information at the Human-Machine Interface." In J. Rasmussen, K. Duncan & J. Leplat (Eds.), *New Technology and Human Error*. London: Wiley.
- Payne, J. W., J. R. Bettman, and E. J. Johnson. (1993). "The Use of Multiple Strategies in Judgment and Choice." In N. J. Castellan, Jr. (Ed.), *Individual and Group Decision Making*. Hillsdale, NJ: Lawrence Erlbaum.
- Payne, S. J. (1984). "Task-Action Grammars." In *Interact '84*, Vol. 1. London: IFIP.
- Perdu, D. and A. H. Levis. (1993). "Requirements Determination Using the Cube Tool Methodology and Petri Nets." *IEEE Transactions on Systems, Man, and Cybernetics*, 23(5), September.
- Pew, R. W. (1988). *Testimony before the U.S. House of Representatives, Committee on Armed Services on the Subject of the USS VINCENNES Downing of Iranian Air Flight 655*. October 6, 1988.
- Pressman, R. S. *Software Engineering: A Practitioner's Approach, 2nd Edition*, McGraw-Hill Book Co., New York, 1987.

- Raphael, T. D. (1991). *Cognitive Flexibility and Innovation in the Problem Definition Process for Tactical Command and Control*. [D-39W-91 (Revised). Prepared for Logistics Research Group, Air Force Human Relations Laboratory, Wright-Patterson AFB, OH 45433-6573.] Falls Church, VA: Mystech Associates, May.
- Rasmussen, J. (1990). "Mental Models and the Control of Action in Complex Environments." In D. Ackerman and M. J. Tauber (Eds.), *Mental Models and Human-Computer Interaction I*. Amsterdam: Elsevier (North-Holland).
- _____. (1986). *Information Processing and Human-Machine Interaction: an Approach to Cognitive Engineering*. New York: North-Holland, 1986.
- _____. (1993). "Deciding and Doing: Decision Making in Natural Contexts." In G. A. Klein, J. Orasanu, R. Calderwood, and C. E. Zsombok (Eds.), *Decision Making in Action: Models and Methods*. Norwood, NJ: Ablex.
- Rasmussen, J. & L. P. Goodstein. (1988). "Information Technology and Work," in Martin Helander (Ed.), *Handbook of Human-Computer Interaction*. Amsterdam: Elsevier.
- Rasmussen, J. and A. M. Pejtersen. (1993). *Mohawc Taxonomy Implications for Design and Evaluation*. Technical Report Risø-M-673(EN). Roskilde, Denmark: Risø National Laboratory, January.
- Rasmussen, J. and K. J. Vicente. (1989). "Coping with Human Errors through System Design: Implications for Ecological Interface Design." *International Journal of Man-Machine Studies*, 31, 517-534.
- Rasmussen, J., A. M. Pejtersen, and K. Schmidt. (1990). *Taxonomy for Cognitive Work Analysis*. Technical Report Risø-M-2871. Roskilde, Denmark: Risø National Laboratory, September.
- Rasmussen, J., A. M. Pejtersen, and L. P. Goodstein. (in press). *Cognitive Systems Engineering*. New York: John Wiley.

- Rastegary, H. and F. J. Landy. (1993). "The Interactions Among Time Urgency, Uncertainty, and Time Pressure." In O. Svenson and A. J. Maule (Eds.), *Time Pressure and Stress in Human Judgment and Decision Making*. New York: Plenum Press, 217-240.
- Ravden, S. J. and G. I. Johnson. (1989). *Evaluating Usability of Human-Computer Interfaces: a Practical Method*. Chichester, UK: Ellis Horwood.
- Robbins, Stephen P. (1990). *Organizational Theory: Structure, Design and Applications*. Englewood Cliffs, NJ: Prentice-Hall.
- Rogers, Y. (1992). "Mental Models and Complex Tasks." In Y. Rogers, A. Rutherford & P. A. Bibby (Eds.), *Models in the Mind: Theory, Perspective and Application*. New York: Academic Press.
- Roth, E.M., K.B. Bennett, and D.D. Woods. (1988). "Human Interaction with an "Intelligent" Machine." In E. Hollnagel, G. Mancini, and D. D. Woods (Eds.), *Cognitive Engineering in Complex Dynamic Worlds*. London: Academic Press.
- Rouse, W. B. (1991). *Design for Success: a Human Centered Approach to Designing Successful Products and Systems*. New York: John Wiley.
- Rouse, W. B., J. A. Cannon-Bowers, and E. Salas. (1992). "The Role of Mental Models in Team Performance in Complex Systems." *IEEE Transactions on Systems, Man, and Cybernetics*, .
- Royce, W. Managing the Development of Large Software Systems. *Proceedings WESCON*, August, 1970.
- Sage, A.P. *Methodology for Large Scale Systems*. McGraw-Hill, New York, NY, 1977
- Sage, A. P., and Palmer, J. D. *Software Systems Engineering*. John Wiley, New York, NY, 1989.

- Salama, A. (1992). "The Use of an Organization's Biography as a Research Method for Investigating Organizational Development." *Management Education & Development*, Autumn, 23(3), 225-233.
- Salvendy, G. (1987). "What We Know and What We Should Know About Human-Computer Interaction: Strategies for Research and Development." In G. Salvendy (Ed.), *Cognitive Engineering in the Design of Human-Computer Interaction. Proc. of the Second International Conference on Human-Computer Interaction, Honolulu, Hawaii, August 10-14, 1987, Vol II*. New York: Elsevier.
- Sanderson, P. M., J. M. Flach, M. A. Buttigieg, and E. J. Casey. (1989). "Object Displays Do Not Always Support Better Integrated Task Performance." *Human Factors*, 31(2), 183-198.
- Senders, J. W. and N. P. Moray. (1991). *Human Error: Cause, Prediction, and Reduction*. Hillsdale, NJ: Lawrence Erlbaum.
- Shanteau, J. (1992). "The Psychology of Experts: An Alternative View." In G. Wright and F. Bolger (Eds.), *Expertise and Decision Support*. New York: Plenum Press.
- Sheridan, T. B. and W. R. Ferrell. (1974). *Man-Machine Systems: Information, Control and Decision Models of Human Performance*. Cambridge, MA: The MIT Press.
- Shneiderman, B. (1992). *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, 2nd Ed. New York: Addison-Wesley.
- Smith, J. B., D. K. Smith, and E. Kupstas. (1993). "Automated Protocol Analysis." *Human-Computer Interaction*, 8, 101-145.
- Smith, S. L. & J. N. Mosier. (1986). *Guidelines for Designing User Interface Software*, Report ESD-TR-86-278. Bedford, MA: The MITRE Corporation, August.

- Staggers, N. and A. F. Norcio. (1993). "Mental Models: Concepts for Human-Computer Interaction Research." *International Journal of Man-Machine Studies*, 38, 587-605.
- Stammers, R. B., M. S. Carey and J. A. Astley. (1990). "Task Analysis," in J. R. Wilson and E. N. Corlett, (Eds.). *Evaluation of Human Work: A Practical Ergonomics Methodology*. New York: Taylor & Francis.
- Svenson, O. and A. J. Maule (Eds.). (1993). *Time Pressure and Stress in Human Judgment and Decision Making*. New York: Plenum Press.
- Tainsh, M. A. (1985). "Job Process Charts and Man-Computer Interaction Within Naval Command Systems." *Ergonomics*, 28, 555-565.
- Thayer, T. A. (1975). "Understanding Software Through Analysis of Empirical Data," *TRW Software Series*, TRW-SS-75-04, May.
- Treu, S. (1992). "Interface Structures: Conceptual, Logical, and Physical Patterns Applicable to Human-Computer Interaction." *International Journal of Man-Machine Studies*, 37, 565-593.
- U.S. Army Research Institute. (1989). *Human Factors Guidelines for Command and Control Systems: Battlefield and Decision Graphics Guidelines. Research Product 89-01*. Fort Leavenworth, KS: U. S. Army Research Institute for the Behavioral and Social Sciences, March.
- Vanderhaegen, F., I. Crevits, S. Debernard, P. Millot. (1994). "Human-Machine Cooperation: Toward an Activity Regulation Assistance for Different Air Traffic Control Levels." *International Journal of Human-Computer Interaction*, 6(1), 65-104.
- Vicente, K. J. and J. Rasmussen. (1992). "Ecological Interface Design: Theoretical Foundations." *IEEE Transactions on Systems, Man and Cybernetics*, 22(4), July/August, 589-606.
- Weick, K. E. and K. H. Roberts. (1993). "Collective Mind in Organizations: Heedful Interrelating on Flight Decks." *Administrative Quarterly*, 38(3),

September, 357-381.

- Welford, A. T. (1976). *Skilled Performance: Perceptual and Motor Skills*. Glenview, IL: Scott, Foresman & Co.
- Wellens, A. R. (1993). "Group Situation Awareness and Distributed Decision Making: From Military to Civilian Applications." In N. J. Castellan, Jr. (Ed.), *Individual and Group Decision Making*. Hillsdale, NJ: Lawrence Erlbaum.
- Wickens, C. D. (1987). "Attention." In P. A. Hancock (Ed.), *Human Factors Psychology*. Amsterdam: North-Holland.
- Wickens, C. D., A. Stokes, B. Barnett, and F. Hyman. (1993). "The Effects of Stress on Pilot Judgment in a MIDIS Simulator." In O. Svenson and A. J. Maule (Eds.), *Time Pressure and Stress in Human Judgment and Decision Making*. New York: Plenum Press, 271-292.
- Williams, K. E., R. E. Reynolds, T. F. Carolan, P. D. Anglin, and L. B. Shrestha. (1989). *An Evaluation of a Methodology for Cognitively Structuring and Adaptively Sequencing Exercise Content for Embedded Training*. Technical Report TR89-035. Orlando, FL: University of Central Florida, Institute for Simulation and Training, December.
- Wilson, J. and D. Rosenberg. (1988). "Rapid Prototyping for User Interface Design." In M. Helander (Ed.), *Handbook of Human-Computer Interaction*. Amsterdam: North-Holland.
- Woods, D. D. (1984). "Visual Momentum: A Concept to Improve the Cognitive Coupling of Person and Computer." *International Journal of Man-Machine Studies*, 21, 229-244.
- _____. (1988). "Coping with Complexity: The Psychology of Human Behavior in Complex Systems." In L. P. Goodstein, H. B. Andersen, and S. E. Olsen (Eds.), *Tasks, Errors, and Mental Models*. London: Taylor & Francis.

- _____. (1991). "The Cognitive Engineering of Problem Representations." In G. R. S. Weir and J. L. Alty (Eds.), *Human-Computer Interaction and Complex Systems*. London: Academic Press.
- Woods, D. D. and E. Hollnagel. (1987). "Mapping Cognitive Demands in Complex Problem-Solving Worlds." *International Journal of Man-Machine Studies*, 26, 257-275.
- Woods, D. D. and E. M. Roth. (1988). "Cognitive Systems Engineering." In M. Helander (Ed.), *Handbook of Human-Computer Interaction*. Amsterdam: Elsevier.
- Wright, P. C. and A. F. Monk. (1989). "Evaluation for Design." In A. Sutcliffe and L. Macaulay (Eds.), *People and Computers V*. Cambridge, UK: Cambridge University Press.
- Wright, P. C. and A. F. Monk. (1991). "A Cost-Effective Evaluation Method for Use by Designers." *International Journal of Man-Machine Studies*, 35, 891-912.
- Zachary, W. W. (1988). "Decision Support Systems: Designing to Extend the Cognitive Limits." In M. Helander (Ed.), *Handbook of Human-Computer Interaction*. Amsterdam: Elsevier.
- Zahniser, R. A. (1993). "Design by Walking Around." *Communications of the ACM*, 36(10), October, 115-123.
- Zhang, W.-R., S.-S. Chen, W. Wang, and R. S. King. (1992). "A Cognitive-Map-Based Approach to the Coordination of Distributed Cooperative Agents." *IEEE Transactions on Systems, Man, and Cybernetics*, 22(1), Jan/Feb, 103-113.

Appendix A

Screen Displays from the DesignPro Workbench

MAIN MENU

Design Projects

Designer

Code

ANALYSIS, DESIGN, & EVALUATION

REQUIREMENTS MODELING



Domain Analysis



Hierarchical Task Analysis



Users



Constraints



Cases



Overall

PROTOTYPING



Display Design



Data & Information Coding



Dialog & Interaction



Implementation



Cases



Overall

PROTOTYPE EVALUATION



Requirements Traceability



Experimental Design



Subjective Testing



Empirical Testing



Cases



Overall

UCI SAMPLING

UCI COTS & SAMPLES CATALOG



Video Samples of UCI Features



Database Development COTS Tools



Multimedia Development COTS Tools



Prototyping COTS Tools



Macintosh Prototyping Tools



General Overview of COTS Tools



Information



Web



Quit

AVAILABLE DOMAIN GUIDANCE

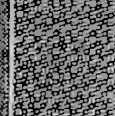
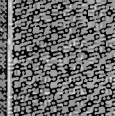
PLEASE SELECT DOMAIN(S) FROM THE LISTING ABOVE BY CLICKING ON THE NAME

SELECTED

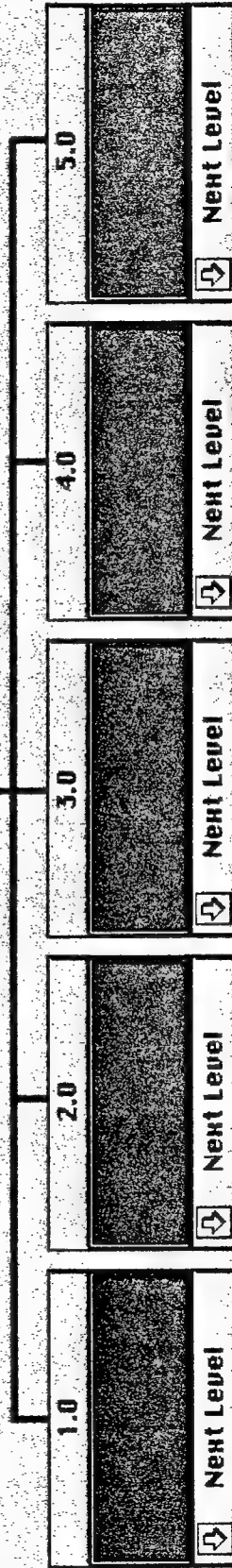
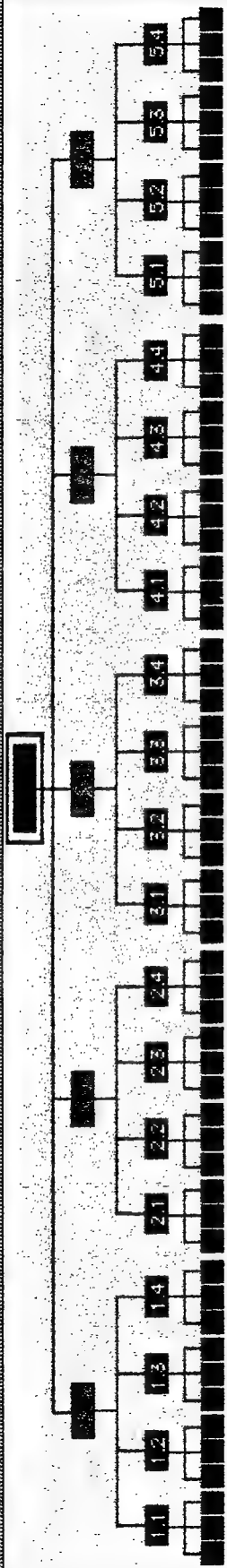
DOMAIN

<input checked="" type="checkbox"/>	CHILD LANGUAGE ACQUISITION
<input type="checkbox"/>	CHILD LANGUAGE DEVELOPMENT
<input type="checkbox"/>	CHILD LANGUAGE INSTRUCTION
<input type="checkbox"/>	CHILD LANGUAGE RESEARCH
<input type="checkbox"/>	CHILD LANGUAGE THEORY
<input type="checkbox"/>	CHILD LANGUAGE USE

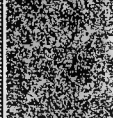
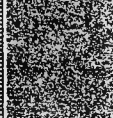
SEARCH RESULTS



HIERARCHICAL TASK ANALYSIS



CLICK ON THE CELL TO ENTER YOUR TEXT



USERS

PLEASE SELECT USED EXPERTISE FROM THE LISTING BELOW BY CLICKING ON THE EXPERIENCE LEVEL

USER EXPERIENCE

EXPERIENCE LEVEL

	HIGH	MEDIUM	LOW
Domain Knowledge			
Task Knowledge			
Computing Experience			
GUI Experience			
Expected Use			
Database Experience			
Spreadsheet Experience			
Programming Experience			

DESCRIPTION FOR

ANNOTATION FOR



Information



Main Menu



Overall



Display Design



Data & Information Coding



Dialog & Interaction



Help



Quit

CONSTRAINTS

Target Platform

IBM Compatible		Macintosh	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Workstation		Other	
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Operating System

IBM Compatible		Macintosh	
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Workstation			
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Graphical User Interface

IBM Compatible		Macintosh	
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Workstation			
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Display Device Color

<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Management

Implementation Team		Schedule	
<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Budget			
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

RAM (MB)

<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Hard Disk Storage (MB)

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>



Information



Main Menu



Overall



Display Design



Data & Information Coding



Dialog & Interaction



Help



Quit

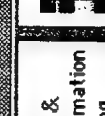
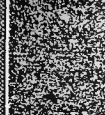
CASES

Available Cases

Tasks Users Constraints

AWACS/E2C	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Fire Control	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Tactical Planning	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Bulletin Board	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Submarine Fire Control	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Emerald	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

DOUBLE CLICK ON THE CASE TO LAUNCH IT



OVERALL REQUIREMENTS STATUS

<div>Task Analysis</div> <div> <div> <div>Characteristics</div> <div>Annotations</div> <div>Importance</div> </div> <div> <div>✓</div> <div>✓</div> <div>✓</div> </div> </div> <div>Not Completed</div>	
<div>Users</div> <div> <div>Characteristics</div> <div>Annotations</div> </div> <div> <div>✓</div> <div>✓</div> </div> <div>Completed</div>	
<div>Constraints</div> <div> <div>Hardware</div> <div>Software</div> <div>Management</div> </div> <div> <div>✓</div> <div>✓</div> <div>✓</div> </div> <div>Not Completed</div>	
<div>Overall Status</div> <div>Proceed/Cannot Proceed to Design & Prototyping</div>	
<div>Recommended Design Steps</div> <div></div>	



Information



Main Menu



Help



Quit

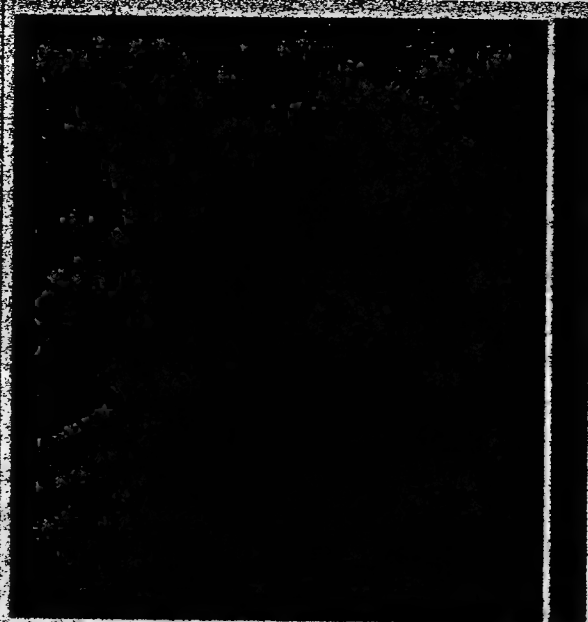
DISPLAY DESIGN

Display Features

Confidence Level

FEATURES:	Yes	No	Weak	Strong
Analog Displays	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Bar Charts	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Column Charts	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Band Charts	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Status Indications	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Pattern Displays	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Trend Displays	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Form Fill-In	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Windows	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Palettes	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Embedded Menus	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Multiple Windows	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Pull-Down Menus	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Pop-Up Menus	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Example

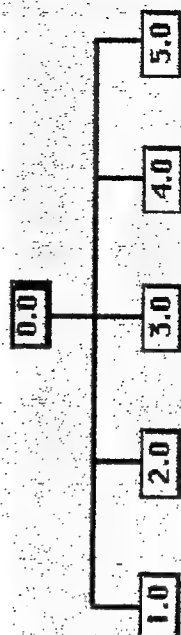


EXAMPLE OF THE FEEDBACK FOR THE FEEDBACK FEEDBACK FEEDBACK

DEFINITION FOR:

RATIONALE FOR:

Diagram



Information



Main Menu



Overall



Data /
Information
Coding



Dialog &
Interaction



Help



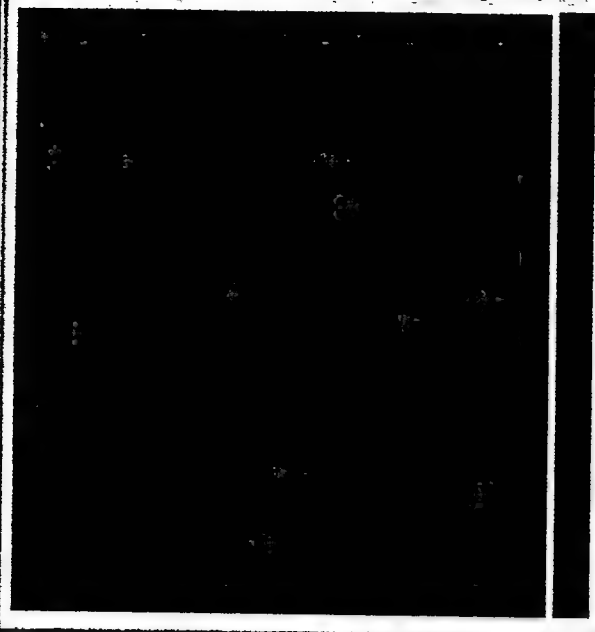
Quit

DIALOG & INTERACTION

Dialog & Interaction Features Inclusion Confidence Level

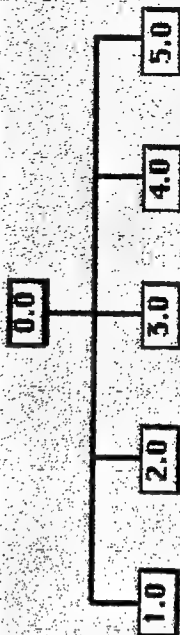
FEATURES	Yes	No	Weak	Strong
Task Automation	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Natural Language	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Hierarchical Menu	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Single Menu	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Command Language	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Hypertext	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
GUI / Iconic	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Command Menu	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Virtual Reality	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Speech	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Example



DEFINITION FOR:

RATIONALE FOR:



Information



Main Menu



Overall



Data /
Information
Coding



Display Design



Help



Quit

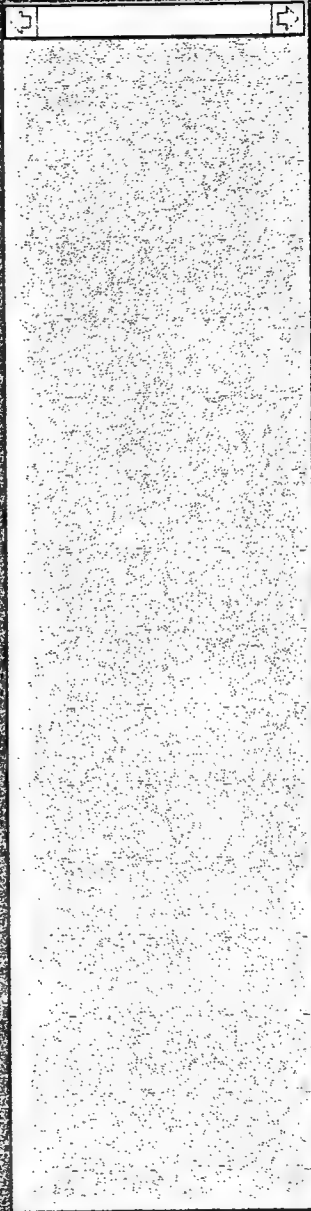


PROTOTYPE IMPLEMENTATION

Prototyping Environment

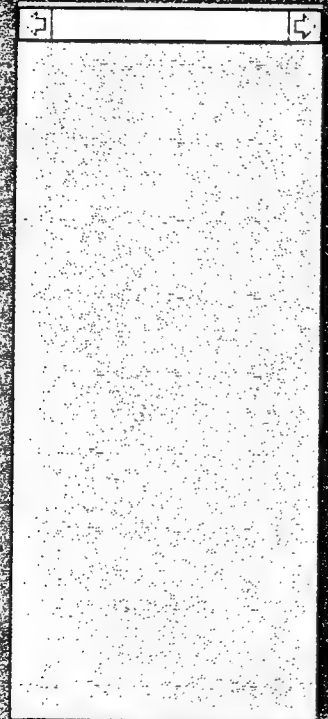


Recommended COTS Prototyping Tools



PROTOTYPING CASES

Available Designs



Available Prototypes

- AWACS / E2C
- FIRE Control
- Tactical Planning
- Counter Terrorism Crisis Management
- Bulletin Board

Available Examples



Information



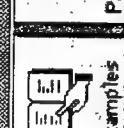
Main Menu



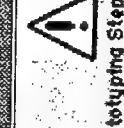
Overall



Examples



Prototyping Step Description






Help



Quit


OVERALL PROTOTYPING STATUS


 **Display Design**

 Data / Information Coding	Completed
 Dialog & interaction	Completed
 Implementation	Completed

 **Information**

 **Main Menu**

 **Help**

 **Quit**



REQUIREMENTS TRACEABILITY

Tasks

Constraints

Users

UCI Features



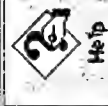
Information



Main Menu



Overall



Help



Quit

EXPERIMENTAL DESIGN

PERFORMANCE MEASUREMENT OPTIONS

	Existing Interface	Enhanced Interface	New System Interface
Subjectively	✓	✓	
Empirically			✓

PERFORMANCE MEASURES OPTIONS

- Time Necessary to Complete the Task
- Time Necessary to Train Users / Operators
- Ease of Use / Training Costs
- Error Rate & Control
- Task Completion Accuracy / Effectiveness
- Situational Adaptability
- Situation Awareness



SUBJECTIVE TESTING

Subjective Testing Checklist



Identify Test Subjects



Identify Testing Logistics



Guidelines



Launch QUIZ



Information



Main Menu



Overall



Help



Quit



Empirical Testing

Empirical Testing Checklist



Identify Test
Objects



Identify Testing
Logistics



Guidelines



Information



Main Menu



Overall







Help

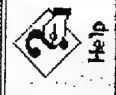


Quit

OVERALL EVALUATION STATUS

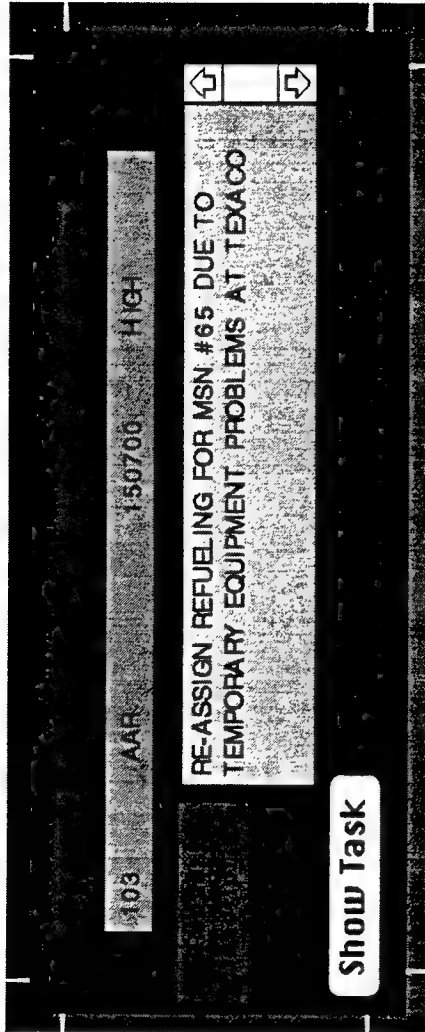
 Requirements Traceability	Not Completed
 Experimental Design	Not Completed
 Subjective Testing	Not Completed
 Empirical Testing	Not Completed

Recommended Evaluation Steps



Appendix B

FLEX “Before” & “After” Screens



Task Notify
FLEX Window (Modified for Case Study)

Task Assigned		Deadline		Request Description		Response Description	
103 AAR 150700				RE-ASSIGN REFUELING FOR MSN 65 DUE TO TEMPORARY EQUIPMENT PROBLEMS AT TEXACO			
Status		Urgency					
REC 150607		HIGH					
PLAN CHANGES							
Assigned DO		Plan Change		Plan Change Description			
AAR 1		2		LINKAGE 02 AT TEXACO UNAVAILABLE 150830/1 51300 DUE TO FUEL SYSTEM PROB RE-ASSIGN MSN 65 NOSEGAY 01 FOR 10 FRE OR 8 MID-EARLY MISSION REFUEL			
CONFLICTS							
Sub Task		Type		Plan Change		Conflict Description	
1003 AAR		1 REFUELING		TANKER ORBIT UNAVAIL		REFUELING UNASSIGNED	

Task Inspector
FLEX Window (Modified for Case Study)

Rqst #	Num A/C	Type A/C	Fuel Req'd	Fuel Pln'd	Not Pln'd	Fuel Type	TOT/TFT	Route Phase	Location	Alt	Num Rcv	AAR MsnID
315	2	F15C	5	0	5	JP4	151235/151245	POST	SHELL	170	4	11
316	2	F15C	5	0	5	JP4	151320/151323	POST	SHELL	170	4	11
101	2	A10A	12	12	0	JP4	151000/151400	PRE	SHELL	170	4	11
101	2	A10A	15	15	0	JP4	151000/151400	MIDEARLY	SHELL	170	4	11
101	2	A10A	15	15	0	JP4	151000/151400	PRE	ESSO	170	1	21
113	1	A10A	10	10	0	JP4	150800/151000	PRE	ESSO	170	1	21
113	1	A10A	11	11	0	JP4	150800/151000	MIDEARLY	ESSO	170	1	21
113	1	A10A	2	2	0	JP4	150800/151000	POST	ESSO	170	1	21
114	1	A10A	10	10	0	JP4	151000/151200	PRE	ESSO	170	1	21

Msn ID	Base	Num A/C	Type A/C	Call Word	Call #	Fuel Type	Location	Alt	Station Alert Time	Offld Avail	Offld Plnd	Offld Rem	Num Rev
11	OBBI	1	KC135R	LINKAGE	00	JP4	SHELL	170	150915/151645	80	70	10	4
21	ABCD	1	KC135R	ROMAN	00	JP4	ESSO	170	150712/151530	84	69	15	3
31	OBBI	1	KC135R	LINKAGE	01	JP4	TEXACO	230	151700/160500	40.35	36	4.35	3
41	OBBI	1	KC135R	LINKAGE	02	JP4	TEXACO	170	150715/151800	46.6	26	20.6	6

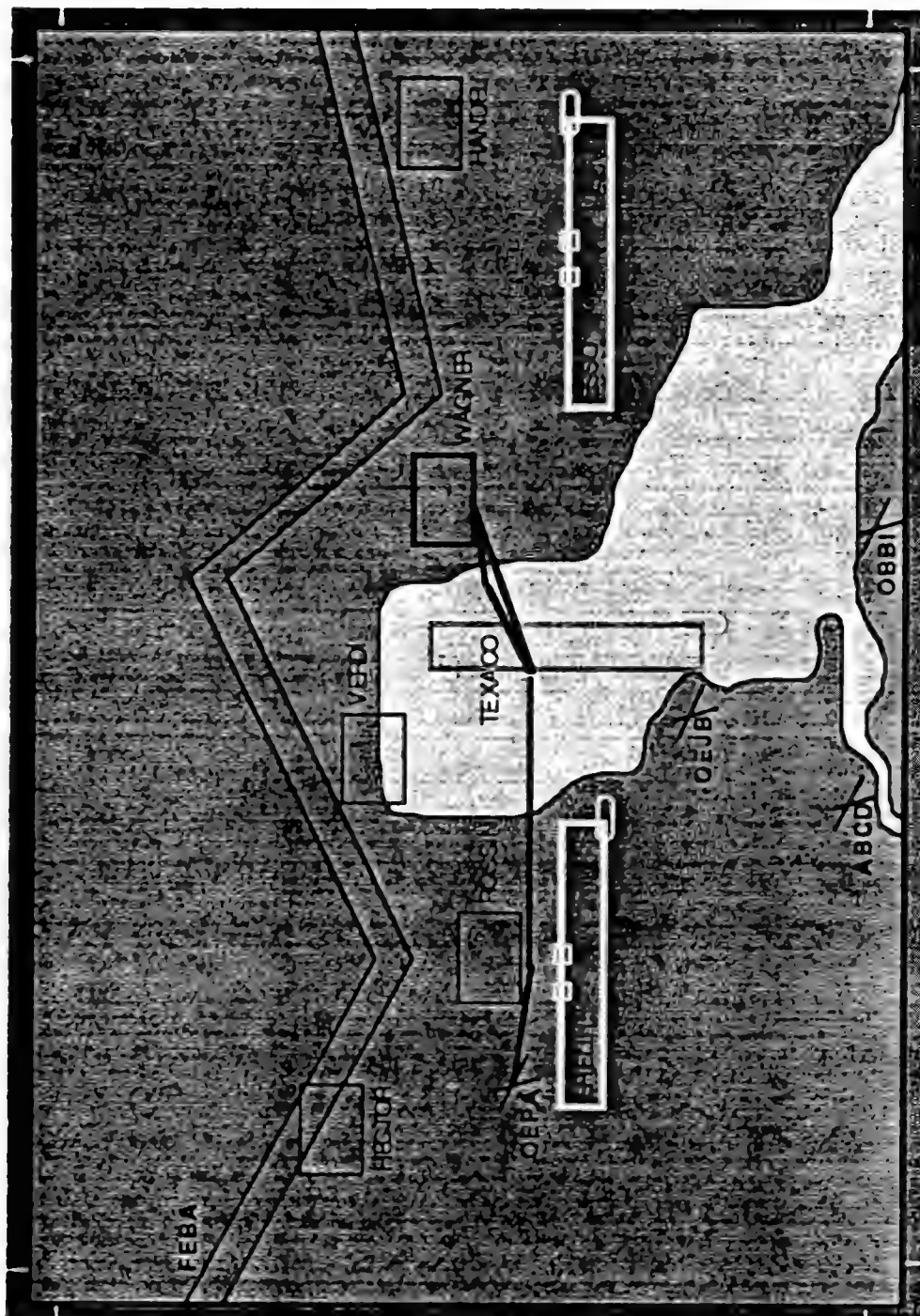
Base	Unit	Type A/C	Fuel Type	Fuel Cap	Total Sorties	Num Tskd
OBBI	25ARS	KC135R	JP4	203	10	3
ABCD	509AR	KC135R	JP4	203	5	1

Location	Start Time	End Time	Min Alt	Max Alt	Tot Rqstd	Fuel Pnd	Tot Off Pnd
ESSO	150712	151800	170	170	69	69	69
SHELL	150915	151645	170	170	70	70	70
TEXACO	150715	151800	170	170	26	26	26
TEXACO	151700	160500	230	230	36	36	36

Tanker Worksheet
FLEX Window (Modified for Case Study)

Msn ID	Location	Alt	FU	CF	Offld	Offld Rem	RCV	Pln Depart	Pln Station	Est Station	Pln Return
11	SHELL	170	JP4	BOM	80	10	4	150832	150915/151645		151726
21	ESSO	170	JP4	BOM	84	15	3	150636	150712/151530		151614
31	TEXACO	230	JP4	BOM	40.35	4.35	3	151631	151700/160500		160527
41	TEXACO	170	JP4	BOM	46.6	20.6	6	150646	150715/151800	150715/150830	151827
41	TEXACO	170	JP4	BOM	46.6	20.6	6	150646	150715/151800	151300/151800	151827

Tanker Status Display Board (SDB)
FLEX Window (Modified for Case Study)

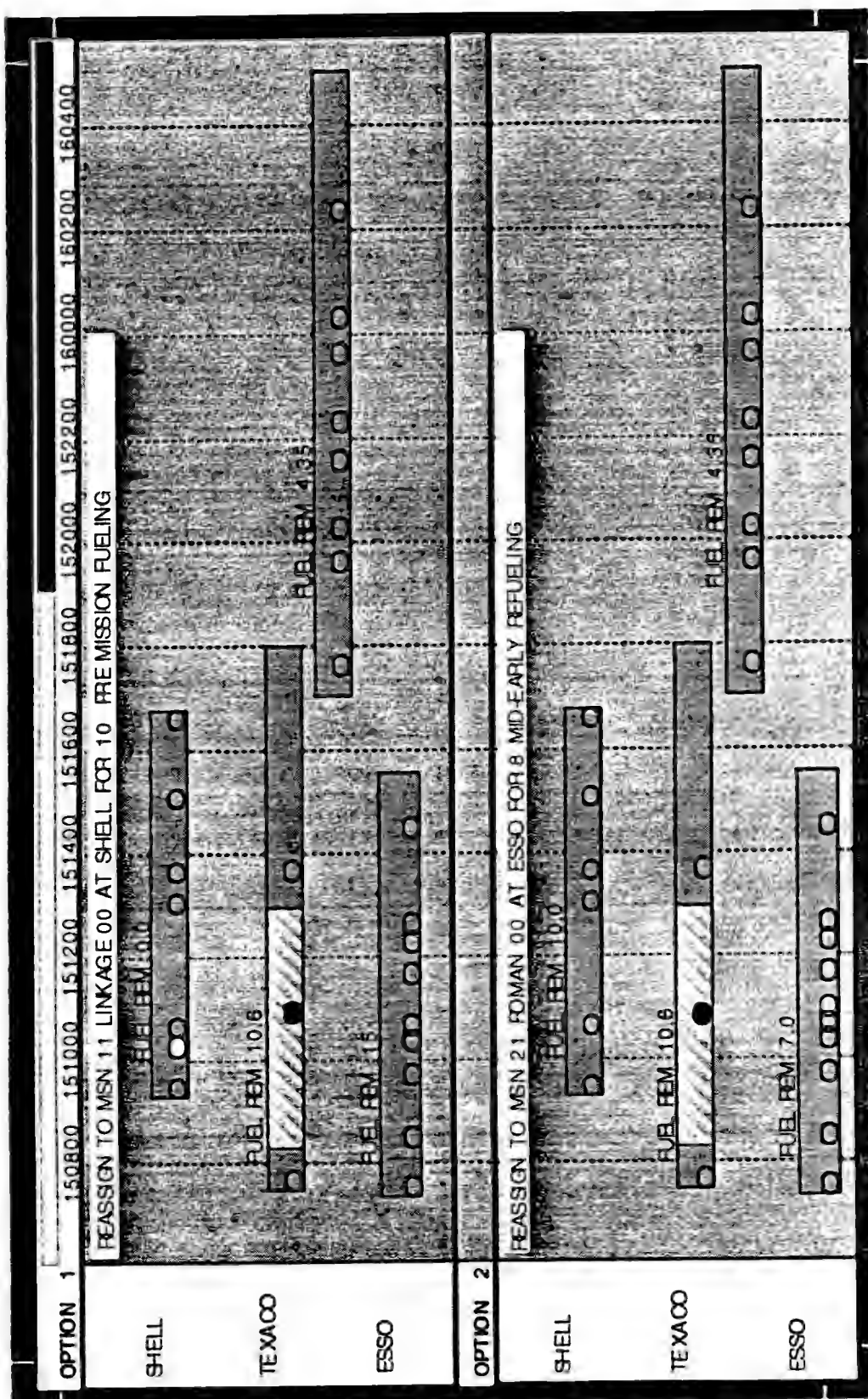


Map Graphic
FLEX Window (Modified for Case Study)

Task 103			
Option ID	Time Created	Rating	Option Description
1	150615	100	REASSIGN TO MSN 11 AT SHELL FOR 10/PRE FUELING
2	150617	100	REASSIGN TO MSN 21 AT ESSO FOR 8/MID-EARLY REFUELING

Replanning Options

FLEX Window (Modified for Case Study)



Option View
 Cognitively Engineered Window
 Proposed Support to Replanning Option Evaluation for Tanker Operations

Click the button next to the BETTER option.

☐ **Option 1:**
RE-ASSIGN MSN 65 FOR PRE-MISSION REFUELING FROM
MSN 11 LINKAGE 00 AT SHELL

☐ **Option 2:**
RE-ASSIGN MSN 65 FOR MID-LATE REFUELING FROM MSN
21 ROMAN 00 AT ESSO

Click Done when you have completed your
selection.

Done

Automated	Data	Collection	Option Select	#1:	Capturing	Decision
-----------	------	------------	---------------	-----	-----------	----------

Subject #: Session #:
 Date: Time:
 Interface:

	Start	Stop	Select	Choice
Trial 1:	9:28:53AM	9:30:50AM	9:31:01AM	1
Trial 2:	9:31:18AM	9:34:04AM	9:34:10AM	2
Trial 3:	9:34:25AM	9:38:32AM	9:38:43AM	2
Trial 4:	9:39:00AM	9:43:12AM	9:43:30AM	2
Trial 5:	9:43:40AM	9:49:54AM	9:49:58AM	2
Trial 6:	9:50:15AM	9:54:48AM	9:54:54AM	2
Trial 7:	9:55:10AM	9:57:01AM	9:57:05AM	1
Trial 8:	9:57:22AM	10:01:32AM	10:01:36AM	1
Trial 9:	10:01:50AM	10:04:03AM	10:04:08AM	1
Trial 10:	10:04:21AM	10:08:29AM	10:08:33AM	1
Trial 11:	10:08:46AM	10:10:51AM	10:10:54AM	2
Trial 12:	10:11:09AM	10:13:49AM	10:13:53AM	1

Done

Subject Card
 Automated Data Collection #2: Speed and Accuracy

Subject #: Session #:
 Date: Time:
 Interface:

Trial 1; 9:29:14 AM: item "Replanning Options"; 9:29:32 AM: item "Option View"; 9:29:46 AM: item "Map Graphic";	Trial 2; 9:31:42 AM: item "Map Graphic"; 9:31:53 AM: item "Marquee"; 9:32:07 AM: item "Replanning Options";
Trial 3; 9:34:54 AM: item "Map Graphic"; 9:35:01 AM: item "Marquee"; 9:35:13 AM: item "Option View";	Trial 4; 9:39:19 AM: item "Marquee"; 9:39:36 AM: item "Option View"; 9:40:03 AM: item "Tanker Worksheet";
Trial 5; 9:44:04 AM: item "Map Graphic"; 9:44:12 AM: item "Marquee"; 9:44:25 AM: item "Option View";	Trial 6; 9:50:33 AM: item "Map Graphic"; 9:50:39 AM: item "Marquee"; 9:50:47 AM: item "Option View";
Trial 7; 9:55:34 AM: item "Map Graphic"; 9:55:40 AM: item "Marquee"; 9:55:54 AM: item "Option View";	Trial 8; 9:57:45 AM: item "Option View"; 9:57:51 AM: item "Map Graphic"; 9:57:59 AM: item "Marquee";
Trial 9; 10:02:15 AM: item "Map Graphic"; 10:02:27 AM: item "Marquee"; 10:02:37 AM: item "Task Inspector";	Trial 10; 10:04:48 AM: item "Map Graphic"; 10:04:56 AM: item "Marquee"; 10:05:12 AM: item "Tanker SDB";
Trial 11; 10:08:59 AM: window "Task Inspector"; 10:09:08 AM: item "Map Graphic"; 10:09:12 AM: item "Marquee";	Trial 12; 10:11:32 AM: item "Map Graphic"; 10:11:43 AM: item "Marquee"; 10:11:50 AM: item "Task Inspector";

Subject Tracker
 Automated Data Collection #3: Process Tracing

***MISSION
OF
AFRL/INFORMATION DIRECTORATE (IF)***

The advancement and application of information systems science and technology for aerospace command and control and its transition to air, space, and ground systems to meet customer needs in the areas of Global Awareness, Dynamic Planning and Execution, and Global Information Exchange is the focus of this AFRL organization. The directorate's areas of investigation include a broad spectrum of information and fusion, communication, collaborative environment and modeling and simulation, defensive information warfare, and intelligent information systems technologies.